CAPTURE, ANALYSIS AND SYNTHESIS OF PHOTOREALISTIC CROWDS

A Thesis
Presented to
The Academic Faculty

by

Matthew Flagg

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
December 2010

CAPTURE, ANALYSIS AND SYNTHESIS OF PHOTOREALISTIC CROWDS

Approved by:

Professor James M. Rehg, Advisor
College of Computing
Georgia Institute of Technology

Professor Irfan Essa
College of Computing
Georgia Institute of Technology

Dr. Kiran Bhat
Industrial Light and Magic
Lucasfilm, Ltd.

Dr. Sing Bing Kang
Interactive Visual Media Group
Microsoft Research

Professor Karen Liu
College of Computing
Georgia Institute of Technology

Date Approved: 14 November 2010

Dedicated to my wife Roxanne,

whose artistic spirit

has motivated my research goals over the years.

And to my parents Jim and Annie Flagg,

for their support and encouragement during my

long journey at Georgia Tech.

ACKNOWLEDGEMENTS

I'd like to thank several individuals who gave me influential advice throughout my training in graduate school. First and foremost, thanks to my advisor, Professor Jim Rehg, for granting me the freedom and means to pursue several research projects outside of mainstream computer graphics and vision. He instilled in me a keen interest in system building and real-time vision systems in particular. Prof. Rehg also taught me how to motivate a technical idea in public speaking and academic writing and most importantly, how to strategically select a research topic. Second, I'd like to thank Professor Irfan Essa, who served on my committee and whose class on Digital Video and Special Effects motivated my early thinking and focus on video-based rendering. Third, I'd like to thank Dr. Sing Bing Kang for making several important suggestions in developing the work on human video textures and crowd tubes. Fourth, I had the pleasure of collaborating closely with Professor Atsushi Nakazawa on human video textures and video object segmentation. He co-developed several key algorithms presented in both Chapters 4 and 5.

I'd also like to thank several labmates who frequently offered work and advice in helping me achieve graduate school milestones. In particular, thanks to Dr. Howard Zhou who started and finished his graduate studies at exactly the same time as me and seemingly worked beside me every step of the way. Howard was a fellow graphics student in a primarily computer vision-focused lab. Thanks also to Jianxin Wu for providing technical advice and much practical help. Charlie Brubaker also gave me technical suggestions which served as the basis for the conflict graph-based representation of crowd tubes.

Finally, I'd like to thank my wife Roxanne, who spent many late nights narrating

research videos, being motion captured in a freezing lab and oil painting with a prototype system. Her gracious cooperation made it possible to meet several tight deadlines and to identify overlooked problems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis explores techniques for synthesizing crowds from imagery. Synthetic photorealistic crowds are desirable for cinematic gaming, special effects and architectural visualization. While motion captured-based techniques for the animation and control of crowds have been well-studied in computer graphics, the resulting control rig sequences require a laborious model-based graphics pipeline to render photorealistic videos of crowds.

Over the past ten years, data-driven techniques for rendering imagery of complex phenomena have become a popular alternative to model-based graphics. This popularity is due in large part to difficulties in constructing the sufficiently-detailed models that are required to achieve photorealism. A dynamic crowd of humans is an extremely challenging example of such phenomena. Example-based synthesis methods such as video textures are an appealing alternative, but current techniques are unable to handle new challenges posed by crowds.

This thesis describes how to synthesize video-based crowds by explicitly segmenting pedestrians from input videos of natural crowds and optimally placing them into an output video while satisfying environmental constraints imposed by the scene. There are three key challenges. First, the crowd layout of segmented videos must satisfy constraints imposed by environmental and crowd obstacles. This thesis addresses four types of environmental constraints: (a) ground planes in the scene which are valid for crowd traversal, such as sidewalks, (b) spatial regions of these planes where crowds may enter and exit the scene, (c) static obstacles, such as mailboxes and walls of a building, and (d) dynamic obstacles such as individuals and groups of

individuals. Second, pedestrians and groups of pedestrians should be segmented from the input video with no artifacts and minimal interaction time. This is challenging in real world scenes due to significant appearance changes while traveling through the scene. Third, segmented pedestrian videos may not have enough frames or the right shape to compose a path from an artist-defined entrance to exit. Plausible temporal transitions between segmented pedestrians are therefore needed but they are difficult to identify and synthesize due to complex self occlusions.

We present a novel algorithm for composing video billboards, represented by *crowd tubes*, to form a crowd while avoiding collisions between static and dynamic obstacles. Crowd tubes are represented in the scene using a temporal sequence of circles planted in the calibrated ground plane. The approach consists of representing crowd tube samples and constraint violations with a conflict graph. The maximal independent set yields a dense crowd composition. We present a prototype system for the capture, analysis, synthesis and control of video-based crowds. Several results demonstrate the system's ability to generate videos of crowds which exhibit a variety of natural behaviors.

CHAPTER I

INTRODUCTION

As a controllable medium, video-realistic crowds are important for creating the illusion of a populated reality in special effects, games and architectural visualization. Plausible crowd imagery should depict a layout-constrained scene with people entering and exiting at specific locations and moving through the environment while responding to dynamic obstacles. Individuals comprising a crowd should also exhibit a variety of behaviors to appear realistic. For example, pedestrians may move in a hurried, goal-directed manner or meander while curiously inspecting other people and points of interest. In current practice, crowd artists animate and render crowd effects via a tedious model-based graphics pipeline.

Model-based crowd animation is made possible at a large scale via simulation, by crafting rules for individual agents, thus freeing the artist from having to painstakingly place every keyframe for each control rig. Most research progress has focused on simulating large numbers of agents which exhibit macro-scale behaviors like lane and vortex forming, which are well-suited for background or distant crowds [97, 77, 49]. Unfortunately, simulated mid-ground and foreground crowds require an expensive and complex production pipeline to convey realism in cloth, body, hair and facial motion, especially when approaching the level of detail exhibited by up-close hero characters. Figure 1 illustrates an example of a complex behavior which would be difficult to achieve with current model-based crowd synthesis techniques.

When carefully planned, a crowd artist or director may successfully compose their desired crowd by filming and building it up person-by-person or one block at a time. See Fig. 2 for a typical hybrid composition of simulated and live action crowds.

Figure 1: Example of realistic behavior: This thesis explores the capture and synthesis of crowds exhibiting complex realistic behaviors such as the one depicted in this figure.

Unfortunately, crowd designers rarely have the resources or ability to edit the layout by gathering blocks of actors for additional takes. For crowds which exhibit detailed behaviors at the individual level, such as tousling one's hair just as another walks by, it is difficult to satisfy behavioral spacetime constraints in addition to collision avoidance, constrained movement from entry to exit, desired levels of crowd density, and clone separation to preserve the perception of crowd variety.

To explore realistic crowd synthesis, we adopt the philosophy of example-based rendering: by directly copying pixels from input imagery of real crowds to output images of synthetic crowds, realism may be attained. This thesis addresses the problem of photo-realistically depicting a variety of natural crowd imagery by composing video clips of individual and group behavior subject to environmental constraints. Our method is capable of synthesizing video-based scenes which display a variety of contextually-plausible behaviors because they are captured from the target output

(A)

(B)

Figure 2: A crowded shot from King Kong (A) Blocks of directed crowds are recorded and composited into the foreground while synthetic crowds make up the background. (B) The purple characters are simulated Massive agents (photos courtesy of Massive Software). Our goal is to synthesize controllable foreground and mid-ground crowds from natural video.

environment. For example, one would expect to see more sight-seeing behaviors in a crowd at the Embarcadero of San Francisco than a major thoroughfare at a college campus between classes.

In contrast with model-based crowd simulation, our approach to crowd synthesis does not involve parameter estimation and tuning to achieve aggregate dynamics. We assume that a plausible crowd may be built up from individual and group behaviors and can support the creation of aggregate effects, such as lane forming, by capturing

Figure 3: The goal of video-based crowd synthesis: A still from the bustling Liverpool train station. Achieving this visual realism with traditional model-based graphics is very difficult. This thesis concerns techniques for capturing and producing crowd animations with this level of visual realism. Photo courtesy of David Sim.

and re-using natural examples of such macro-scale behaviors. Using a prototype system for video-based crowd synthesis, we demonstrate how a crowd artist can execute our novel example-based process to realizing a hand-drawn sketch of a crowd. In building our system, we identified and addressed several challenges to video-based crowd synthesis.

1.1 Challenges

Video-based crowd synthesis must address two categories of technical challenges: (1) Synthesis is a constrained layout problem and constraints must be represented and enforced while preserving variety, (2) Video-based crowds must be segmented and re-used while keeping video objects intact and smooth in motion.

Layout constraints: A video of a crowded scene may exhibit large variations

in the density, appearance and motion of its pedestrians (see Fig. 3). The presence of such variations are important to crowd plausibility yet are difficult to achieve due to an abundance of layout constraints. The process of copying input pixels to the output must satisfy environmental constraints imposed by the scene. For example, crowds should not float in mid-air above the ground planes, walk in traffic or through walls of buildings. A crowded scene exhibits static and dynamic obstacles such as mailboxes and pedestrians which further constrain the layout of animated video clips.

Video segmentation and re-animation: In addition to satisfying layout constraints, a video-based crowd exhibiting good crowd variety requires the ability to clip out and re-use recorded clips of crowd behavior in a perceptually-convincing manner. Straightforward techniques for video segmentation, such as background subtraction, are usually insufficient as they do not enforce temporal coherence to prevent pieces of foreground, such as heads and arms of pedestrians, from noticeably scintillating during playback as they oscillate between inclusion and exclusion from the video object. Following segmentation, crowd synthesis must copy and place segmented clips of behavior back into the scene. Current techniques for video re-synthesis, such as video textures and video sprites [87], are incapable of identifying transition points in articulated pedestrian video.

In summary, video-based crowd synthesis faces three key challenges:

- The placement of segmented video clips exhibiting desired behaviors must satisfy spatio-temporal constraints imposed by the environment. This thesis addresses four types of environmental constraints: (a) ground planes in the scene which are valid for crowd traversal, such as sidewalks, (b) spatial regions of these planes where crowds may enter and exit the scene, (c) static obstacles, such as mailboxes and walls of a building, and (d) dynamic obstacles such as individuals and groups of individuals.

- Pedestrians and groups of pedestrians should be segmented from the input video with no artifacts and minimal interaction time. This is challenging in real world scenes due to significant appearance changes while traveling through the scene and shape changes from non-rigid deformations such as cloth crumpling and hair tousling.

- Input video clips of crowd behaviors may not have enough disoccluded frames or the right shape to compose a path from an artist-defined entrance to exit. Plausible temporal transitions between segmented clips of pedestrians are therefore needed to extend the input video via temporal concatenation of frames. Transitions are difficult to identify and synthesize due to complex self occlusions, articulations and the secondary motion of clothing and hair.

1.2 Our Approach

We hypothesize that videos of crowds exhibiting controllable realistic behaviors can be constructed from video clips of individuals and groups. Contrary to previous work in crowd synthesis, which focused on macro-scale crowd behavior via model-based simulation, our focus is on realistic micro-scale behavior and appearance. We approach the problem by aggregating and laying out examples of behavior captured from natural crowds. Within this approach, we explore and address the following research questions:

- Given the layout of the scene, how can we synthesize crowds that respect the appropriate environmental constraints?

- How can we control captured behaviors, their density and duration in synthetic crowd video?

Thesis Statement: Videos of crowds exhibiting realistic controllable behaviors may be synthesized by copying and composing pixels represented by crowd tubes.

Figure 4: Scene before and after video-based crowd synthesis

To address the primary challenge of satisfying constraints on layout and behavior, we cast the video synthesis problem as a constrained 3D layout problem (see Fig. 4). Elements of a crowd including individuals, small groups, and inanimate objects occupy tube-like volumes in the output video volume as illustrated in Fig. 5. By copying, translating, scaling, concatenating and pasting captured video volumes into the output, we crowd the output one instance at a time. After filling the output volume with a seed set of crowd elements, we select a constraint-satisfying subset of elements to make visible during rendering. No pair of elements in the subset should

Figure 5: Crowd tubes in the output video volume. The temporal dimension is orthogonal to the document you are reading. All background pixels have been removed except those in the last frame to show the output video volume's internal crowd tube structure.

collide with each other at any point in time. Replicated elements should be spatially separated to preserve the perception of crowd variety [73]. Each element should enter and exit the scene in a user-specified way. Thus, the problem of placing video volumes into a larger one is reduced to deciding which set of elements in a seed set should be made visible during rendering.

1.2.1 Crowd Tubes

We introduce *crowd tubes*, our representation of the visual elements of a crowd as a sub-volume in the output video volume coupled with a trajectory of 3D shape proxies planted in a calibrated ground plane. Each shape serves as a proxy for a video object's ground plane occupancy for one frame. For example, when swept from the first frame to last, a trajectory of circles representing the time-varying ground plane occupancy forms a well-defined tube in the top-down rectified video volume, like a stack of coins. This enables simple detection of collision and spatial separation constraint violations.

Tighter fitting proxy shapes, such as ellipses or bounded volume hierarchies, could also be used to construct crowd tubes with more complicated geometry to achieve greater constraint violation detection accuracy and increased density.

Crowd tubes are generated by concatenating ground plane trajectories at transition points, thus forming a rigid path for 3D video billboards to follow. The result is a 3D composition of numerous video billboards placed in front of a static background video billboard. As opposed to a 2D layer-based composition, a 3D layout enables straight-forward visible surface determination and correct perspectivity. Figure 5 illustrates a set of crowd tubes composed to form a crowd.

We represent each crowd tube as a node in a conflict graph and insert edges for binary constraint violations including collision and clone separation in order to identify a constraint-satisfying subset for final animation and rendering. The synthetic crowd population is equivalent to the number of selected nodes in the conflict graph. Binary constraints are satisfied when each pair of selected nodes share no edge in the conflict graph. An independent set satisfies this condition and the maximal independent set yields the most populated crowd result achievable from a sample set.

Extension to rendered crowds: In this thesis, crowd tubes are employed to represent, layout and compose recorded clips of natural crowd behavior. However, crowd tubes may be used with sources of video other than cameras. Crowd imagery generated with more traditional model-based software can also serve as the basis of crowd tubes. In this manner, traditional crowds which are expensive to produce in current visual effects pipelines may be re-used to generate novel crowds. For example, a crowd video produced over months using Massive software [3], may require a last minute modification. Such modifications would likely require tweaks to behavioral rules for agents, re-simulation and rendering. Using crowd tubes, pre-rendered behavioral clips may be directly applied to modify crowd effects.

Assumptions: While the crowd tube representation does not expect input pixels

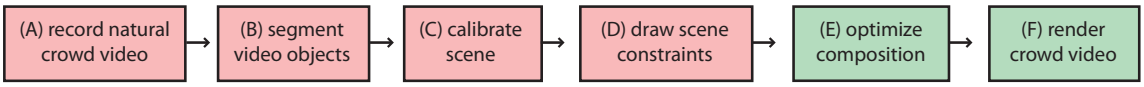


Figure 6: Video-based crowd synthesis pipeline. Steps shown in red require user interaction and green steps are fully automatic.

to come from a video camera, there exist some limiting assumptions. First, crowd tubes are constructed using trajectories with respect to a single ground plane. For example, it cannot represent a pedestrian walking up a tiered staircase. Second, the proxy shape used to generate a crowd tube assumes that the ground plane is occupied by that shape for each frame. Thus, a skateboarder jumping over a park bench cannot be represented with our current crowd tube design. However, the basic representation may potentially be extended to provide for such crowd behaviors.

1.3 System and Dataset

To investigate our approach to video-based crowd synthesis, we built a prototype system and used it to compose several crowded videos exhibiting controlled behaviors and layouts. Using the process illustrated in Fig. 6, an artist can successfully author a photo-realistic crowd video. Chapter 3 describes each of these steps in detail. The next few paragraphs summarize the process.

First, the artist video records a natural crowd by panning, tilting and zooming into the scene to capture short clips of individual behaviors. Next, the artist interactively segments desired video objects by annotating foreground-background regions for keyframes throughout the sequence. Chapter 4 describes the segmentation problem and a novel solution. Following capture and segmentation, the artist interactively calibrates a zoomed-out video clip to extract a 3D ground plane as the basis for video billboard layout. The user manually specifies an output view to top-down view homography to transform trajectories of crowd elements into "sidewalk" coordinates. The artist also specifies a homography mapping each clip of individual behavior to

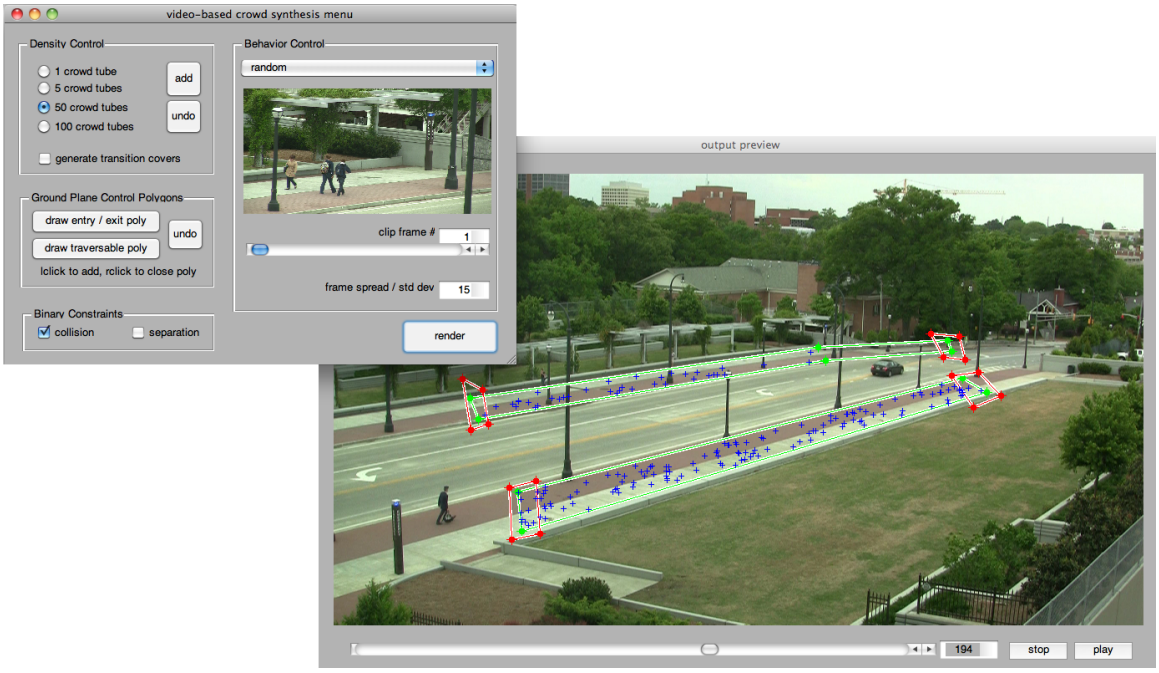


Figure 7: Crowd authoring interface. A crowd artist can choose from a database of input clips and a set of constraints in the left-hand menu. The constraints and crowd tubes are added and manipulated to choreograph a crowd in the output view on the right.

the background output clip. We compute a 2D ground plane position for each frame of each input clip and use the input-output and output-sidewalk homographies to construct ground plane trajectories.

Following calibration, the artist specifies traversable regions and entry and exit regions by drawing polygons on the ground plane using our crowd authoring interface (see Fig. 7). In-polygon testing enables acceptance and rejection of individual crowd tubes. The artist generates a seed set of crowd tubes by clicking on the ground plane to "spray" a set of crowd tubes of user-specified density into the output view. Each spray is controlled in terms of input clip choice, keyframe of the input clip and keyframe in the output clip. Keyframe control enables tight behavior control. The spray interaction technique allows for loose density control. Following crowd authorship, the system constructs a conflict graph, computes a large independent set using Luby's algorithm [71] and renders the selected nodes as a 3D composition of

Figure 8: Selected frames from gallery of four synthetic crowds

animated video billboards.

To validate the approach, we composed several crowds using data captured from four outdoor scenes (Fig. 8). In three of the scenes, the video results depict examples of natural crowds where natural is defined as randomly populated without explicit choreography. In the fourth scene, an actress was recorded performing specific umbrella-opening behaviors wearing various costumes for the purpose of authoring a behavior-choreographed set of crowds. The results demonstrate that our prototype system can be used to author a range of crowd types in a variety of settings. In addition to producing video output results, we conducted an informal user evaluation of the system. Three users deemed to have experience with advanced video editing software used our system to realize a hand-drawn sketch of a crowd. We report their comments and show before and after renderings of their crowd videos. Finally, we provide the segmented video clips and calibrated scenes for further experimentation and crowd authoring by other researchers and artists. Our system and dataset is available on the project website.

1.4 Contributions

This thesis presents five contributions:

- Crowd tubes, a novel representation of the visual elements of a crowd for synthesizing crowd video including individuals, groups of individuals and inanimate objects, coupled with temporally-varying 3D shape proxies, which enable placement of crowd elements in a 3D scene while satisfying constraints from the scene.

- Formulation of a constraint satisfaction problem based on crowd tubes, which enables the synthesis of video-based crowds with properties such as collision avoidance and spatial separation of clones which are necessary for realism.

- The first experimental results for video-based crowd synthesis from video.

- A method for interactively segmenting crowd tubes using temporally-coherent graphcuts and a novel ground-truth database for evaluating segmentation-based tracking algorithms.

- A method for simultaneously capturing marker and video data of human movement and constructing a video graph, which enables the creation of human video textures.

These results extend previous techniques in computer graphics for crowd synthesis and demonstrate the feasibility of video-based crowd composition.

1.4.1 Document Organized by Artifact Space

The rest of the document is organized into two major categories of challenges and their corresponding visual artifacts: (a) constrained layout and composition of crowd behavior at the *video object level of detail* and (b) segmentation and re-synthesis

Table 1: Visual artifacts facing video-based crowd synthesis: Chapter 3 addresses constrained layout of crowd video objects, the first category of technical challenge. Chapters 4 and 5 describe our approach to segmentation and re-synthesis of recorded crowd video, the second category of challenge.

Artifact	**Cause**	**Effect**	**Chapter**
Misplacement	Replicating clips in output video without awareness of scene layout	Crowd elements appear to float above ground or other violations of physics	3
Incorrect perspective	Copying and pasting video clips into a perspective projected scene without scaling	Lacks expected foreshortening, lacks parallax motion, elements appear to float above ground	3
Collisions	Crowd elements occupy the same space on the ground plane	Portions of a crowd element simultaneously occlude and are occluded by another	3
Clone proximity	Copying a crowd element and pasting it in close proximity to itself.	Perception of crowd variety is damaged [73]	3
Segmentation error	Video clips exhibiting large overlap in figure-ground color distributions, severe shape changes between frames and large interframe motion	Composited crowd elements miss portions of the body and scintillate in a visually noticeable manner	4
Transition popping	Video clips of crowd elements are concatenated at transition points exhibiting noticeable differences in shape, color and dynamics	Discontinuous dynamics (popping animation) and appearance mismatch (misregistration) is visually distracting	5

of recorded crowd behavior at the *pixel level of detail*. Table 1 lists a set of visual artifacts which are caused by the key challenges if left unaddressed.

Accordingly, this dissertation is organized into five additional chapters detailing: (2) background material on crowd synthesis, (3) video-based crowd synthesis as a constrained video billboard layout problem, (4) general video object segmentation using temporally-coherent local graphcuts, (5) human video textures with a focus on synthesis of plausible transitions from joint video and motion capture data and (6) conclusions and future work.

The work presented in this thesis reflects the culmination of three independent cuts across the problem space, characterized by visual artifacts. The algorithms and prototype systems were developed and evaluated to address each challenge separately and not altogether as a whole. More specifically, we did not use the video object segmentation technique described in Chapter 4 in the constrained layout system and pipeline described in Chapter 3. Rather, we employed the "Rotobrush"

tool in Adobe After Effects CS5, a commercial video object segmentation software package. Furthermore, the transition synthesis algorithm described in Chapter 5 on human video textures was not utilized to generate the synthesized crowd results of Chapter 3. Instead, we used basic cut transitions which directly concatenate video segments without smoothly transitioning using synthetic transition frames. We leave the challenging work of integrating these pieces into a unified system as future work.

CHAPTER II

BACKGROUND

Traditional crowd synthesis in computer graphics employs techniques for modeling crowd behavior both procedurally and from data. In contrast with our method, these techniques output joint angles and trajectories instead of pixels. The film industry frequently uses Massive [3, 44], a popular agent-based crowd simulation system in which each agent plans their own motion individually. Sophisticated agent-based methods model behavioral dynamics, or how to respond to stimulus, cognitive aspects such as terrain knowledge and learning, and pedestrian visibility for path planning [37, 90]. Many researchers have also tried simpler agent-based models which avoid cognitive modeling [13, 43, 42, 70, 75, 93]. In this chapter, we review particle and image-based crowds as background for the proposed video-based technique.

2.1 Particle-based crowds

Early work on flocking and herding treated crowd simulation as an elaboration of particle systems, with the boid flock model [83] being a seminal example. Particles in the boid flock model are represented with oriented objects, such as birds, which exhibit complex behaviors governed by internal bird state and the external flock state. This approach was an alternative to scripting the paths of each flock component individually. The boid flocking model is based on the following three heuristic rules in order of decreasing precedence: (1) collision avoidance, (2) velocity matching: match velocity with nearby flockmates and (3) flock centering.

In contrast, the proposed approach manipulates a set of video objects and their 2D ground plane trajectories to animate a video-based crowd. Thus, our crowd model is closer to the alternative method to Reynold's boid which consists of scripting

the paths of each crowd component individually. Our trajectories are data-driven, however. A fixed family of trajectories is the set of possible crowd tubes which may be plausibly generated from a single crowd tube. A family of trajectories is generated via translation about the ground plane and temporal re-sequencing.

It is not clear how much shape change a crowd tube could undergo before damaging plausibility. Consider the appearance of a pedestrian recorded while walking in a direction that is fronto-parallel to the camera's image plane. How can we re-animate this example to walk away from the camera in the depth direction? The back is not visible and the motion of the feet is incorrect. Resulting footskate artifacts could potentially be hidden in a dense crowd, but trajectories which disagree with body orientation seen in the head and upper torso may be noticeable. For example, a forward-facing pedestrian moving sideways may give away its synthetic nature, even if the head is solely visible.

Particle-based crowds rest upon the assumption that control rigs can be animated to move along arbitrarily shaped ground tracks. Potential fields used in [97] have been applied to achieve effects such as lane forming and bottle-necking. Massive animates lower level control rigs by blending motion capture data. A related motion capture-driven technique for synthesis of crowds is the method of [94]. A traditional motion graph [57] is used with probabilistic maps to build a crowd of goal-directed individuals one at a time in a greedy fashion. The method makes potentially severe updates to joint angles in order to satisfy hard spacetime constraints.

Crowd patches [103], an extension of motion patches [63], efficiently simulates large crowds for real-time applications by tiling blocks of pre-computed local crowd simulation. A crowd patch is space-time block of pre-computed trajectories which are cyclic over a constant period. By interconnecting neighboring patches with compatible trajectory entry and exit points, animated objects appear to seamlessly cross the limits of a patch to a neighbor during looping playback. We originally considered

adopting a video-based crowd patch approach to densely packing the video volume. However, the technique would require a tedious planning and capture process in order to design scene-dependent crowd patches, direct pedestrians to carefully satisfy patch boundary conditions at the right point in space and time, and synthesize transitions needed in the beginning and end of each period.

In contrast to constructing tile-like examples of crowds for a given spatial area, our approach more closely resembles the example-based methods of [65, 62] to simulate crowds. Lerner et. al. use a set of trajectories extracted manually from crowd video to control autonomous agents. Agent trajectories are incrementally synthesized by considering spatio-temporal relationships with nearby agents and searching for similar scenarios in a trajectory database. They employ locally weighted linear regression to model an agent's speed and moving direction as a function of the motion of nearby agents, its own motion and environment features. The dynamic model is applied to animate traditional model-based avatars. Our system takes a more extreme example-based approach to attain photorealism by re-using captured trajectories and pixels (represented by crowd tubes) with modifications limited to translation and re-arrangement in time.

2.2 Image-based crowds

A number of researchers have exploited image-based proxy representations of pedestrians for rendering crowds of thousands efficiently [96, 52]. These methods enable fast drawing by instancing billboards mapped with textures obtained by rendering more detailed 3D models. Our approach trades such rendering efficiency for visual quality by instancing billboards texture mapped with real video.

Lalonde et. al. describe a closely related system for inserting new objects into existing photographs by sampling from a large image-based object library [61]. Their key idea is to search a database of over 13,000 objects for examples with lighting that

is consistent with the scene. Like our system, theirs has a need to estimate the ground plane for perspectively-correct composition. Another similarity to our problem is the need for high quality segmentation and matting. Our goal for video-based crowd composition faces layout challenges which they do not address. Their system's ability to produce plausible photographs gives us inspiration the crowd video domain.

CHAPTER III

VIDEO-BASED CROWD SYNTHESIS

Over the past ten years, data-driven techniques for rendering imagery of complex phenomena have become a popular alternative to model-based graphics. This popularity is due in large part to difficulties in constructing sufficiently detailed models required to achieve photorealism. A dynamic crowd of humans is an extremely challenging example of such phenomena. Crowd synthesis must satisfy an abundance of constraints on motion and appearance to maintain plausibility. The motion and appearance of a crowd are influenced by the context of their environment.

A crowded scene often includes a flow of human traffic with multiple motion directions within a given spatial area. Motions are constrained by entrances and exits into the scene with boundaries consisting of physical obstacles such as buildings, parked cars and streets with vehicle traffic. Crowd motion itself constrains the flow, especially when the crowd is very dense. Imagine the start of the New York marathon. Runners have a clear goal dictating their direction, yet those in the middle or back of the crowd must wait until there is enough room to start running. It is therefore important to incorporate knowledge of the scene and its pedestrians for constrained video-based crowd synthesis.

Crowd appearance should likewise be informed by environmental context. First and foremost, perspective and positioning of crowds are typically dictated by ground planes in the scene. The Liverpool train station shown in Fig. 3 has two ground planes where crowds lie. The ground plane on the first floor exhibits more perspective foreshortening than the plane on the second floor due to the camera's position relative to the planes. Second, crowd appearance depends on scene lighting. Figure 3 displays

Figure 9: Composition of crowd elements: A video-based crowd is composed using video clips of individuals, groups of individuals and inanimate objects. This chapter describes how to control the layout, density, behavior and duration of a video-based crowd.

both light and dark regions of the scene. Crowds in the center are generally brighter than crowds on the sides.

In this chapter, we explore techniques which leverage the environmental context to synthesize plausible crowd videos from data. We raise the following specific research questions:

- Given a video of a sparsely crowded scene, how can we synthesize a plausible video of the same scene populated with crowds of artist-controlled density and duration?
- Can we control captured behaviors of individuals and groups of pedestrians in the synthetic crowd?

This chapter explores these questions and presents a novel approach and details of our prototype system.

3.1 Problem Statement

Given an input video of a sparsely crowded scene recorded by a camera with a fixed center of projection, synthesize a plausible video of the same scene populated with instances of recorded video objects of artist-controlled duration, density and behavior.

Assume the input contains the following video segments obtained by panning, tilting and zooming as captured by a tripod mounted camera:

- V, a zoomed out view of the scene to be used as a background plate during crowd synthesis.
- V_i, a set of zoomed in views indexed by variable i, which contain fixed view footage of individuals and groups of individuals targeted by the recorder. V_i serves as a "palette" of natural behaviors which comprise the video-based crowd medium.

A plausible crowd video must satisfy constraints on the appearance and motion of crowd elements exhibited in V_i:

- Elements must rest on the ground plane in the scene and exhibit correct perspective.
- Elements can only occupy traversable sections of the ground plane.
- Elements must enter and exit from semantically correct areas of the scene (i.e. they must not appear and disappear in free space).
- No pair of elements should collide (occupy the same space on the ground plane).
- No pair of elements cloned from the same V_i should be in close proximity. This perceptual constraint is motivated by recent findings from a study of the perception of crowd variety [73].

Figure 10 illustrates the problem setup including an input view, constraints necessary for plausibility, and an example output view.

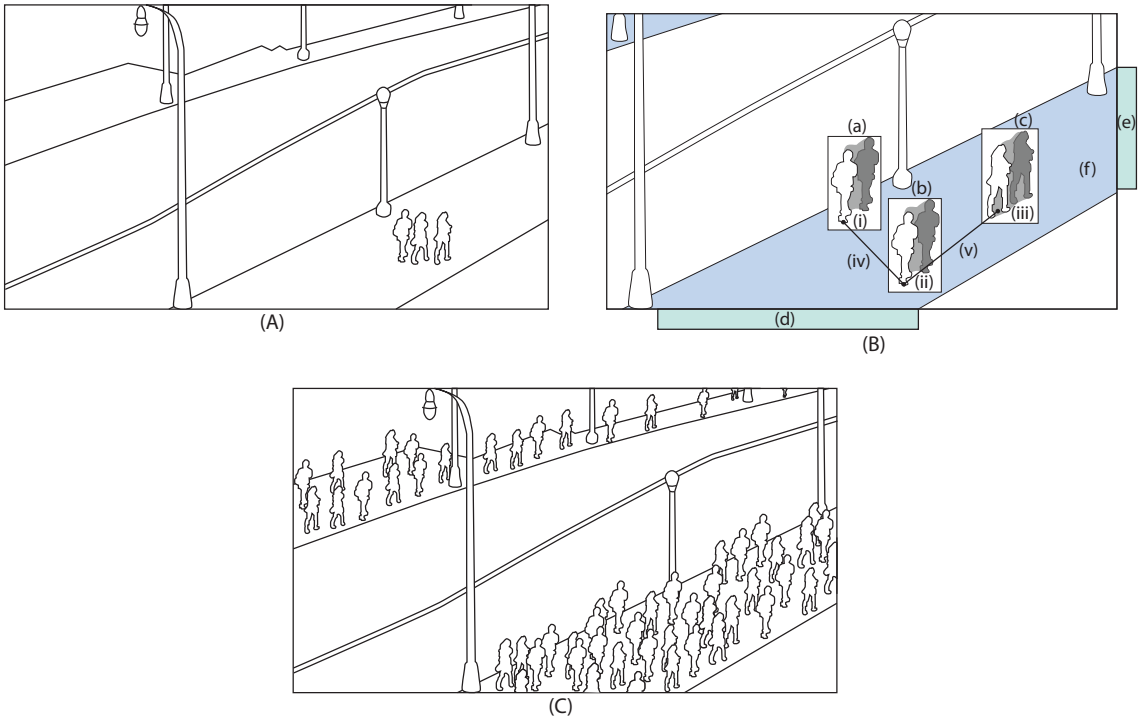


Figure 10: Input, Constraints and Output. (A) **Input:** Sparse crowd video in the target scene. (B) **Constraints:** Segmented video clips are composed in the output video subject to a minimal separation distance (v) between ground locations (ii, iii) for collision avoidance, a minimal separation distance (iv) between ground plane locations (i, ii) for clones, entry and exit regions of interest (d and e) and pedestrian traversability regions (f). Transition visibility is approximated by the distance between a transition frame of one crowd tube and a non-transition frame of another crowd tube (e.g. distance (v) when (c) is transitioning and (b) is not). (C) **Output:** A crowd video which satisfies layout constraints.

3.2 Solution Approach

We approach the problem by treating it as a rigid trajectory layout problem. Each V_i is segmented and its ground plane track, illustrated in Fig. 11, is used to animate an alpha-matted video texture billboard. A video texture billboard or video sprite is a camera-facing 3D plane texture mapped with a video texture [87]. The ground plane track is automatically computed from a segmented V_i by using the foreground centroid's x coordinate and the segmented object's bottom-most y coordinate. Interactive calibration yields a set of homographies and ground plane parameters which are

Figure 11: Crowd tube trajectory: The projected crowd tube trajectory of the man in a blue suit. The image is a zoomed-in region of an *Embarcadero* result.

used to transform coordinates between input, output, top-down "sidewalk" coordinates, and 3D world coordinates planted in the ground plane. Section 3.3.3 describes the calibration steps in detail.

A crowd artist interactively specifies unary constraints on rigid trajectory samples by drawing a set of polygons in the output coordinate system. Polygons represent traversable regions, entry-exit regions and obstacles such as mailboxes. In-polygon testing on the trajectory samples in sidewalk coordinates is performed to decide acceptance or rejection.

Following unary constraint satisfaction, binary constraint satisfaction is performed by computing an independent set on a conflict graph as described in Section 3.2.4. The resulting set of trajectories is laid out as a 3D composition of numerous video billboards placed in front of a static background video billboard. As opposed to a 2D layer-based composition, a 3D layout enables easy visible surface determination and correct perspectivity. Figure 5 illustrates a set of crowd tubes composed to form a crowd.

3.2.1 Crowd Tubes for Constrained Video Billboard Composition

We introduce *crowd tubes*, our representation of video objects as a sub-volume in the output video volume coupled with a trajectory of proxy shapes planted in a

calibrated ground plane. Each shape serves as a proxy for a video object's ground plane occupancy for one frame. We assume that pedestrians are always in contact with a single ground plane and in our experiments, we represent ground occupancy with a circle for simplicity. When swept from the first frame to last, each circle trajectory forms a well-defined tube in the top-down rectified video volume (Fig. 12), like a stack of coins.

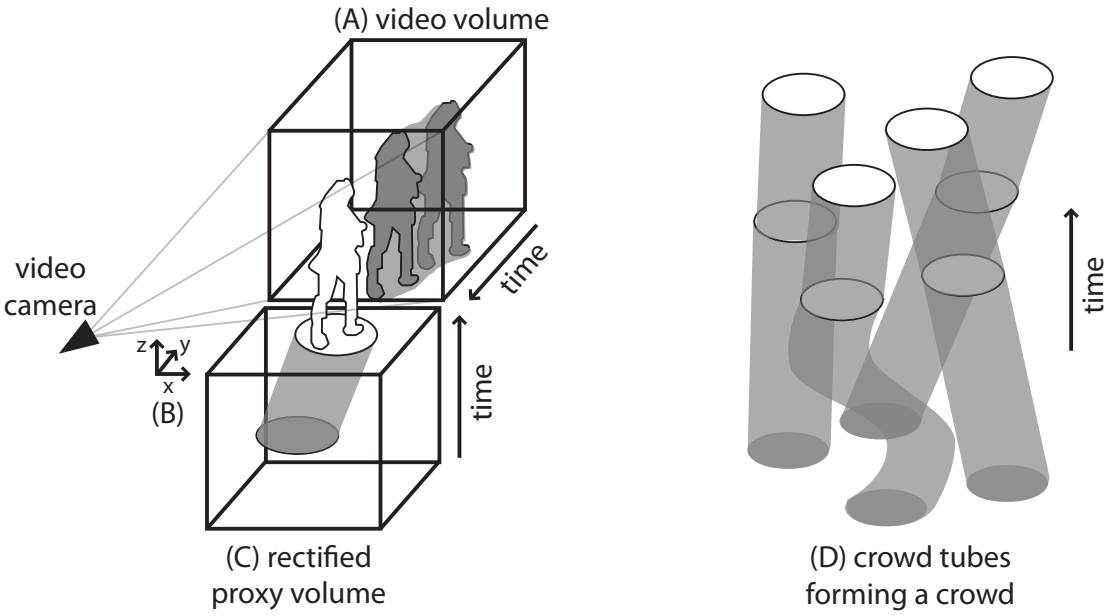


Figure 12: Rectified Proxy Volume. Segmented video clips require 3D scene information to satisfy constraints such as collision avoidance. (A) The video volume of a segmented pedestrian carves out a tube-like shape. (B) The pedestrian's ground track in the input video is converted to world coordinates. (C) A circle planted in the ground plane for each captured frame serves as a proxy for the pedestrian's ground occupancy. Over time, this produces a proxy volume corresponding to the video volume. By rectifying each frame of the video volume to that of a top-down view, a ground plane-over-time representation enables constraint enforcement (collision avoidance, spatial separation of clones, etc.). We define a *crowd tube* as a proxy volume coupled to its corresponding video volume and use it for constraint violation detection and video billboard animation and rendering. (D) A collection of non-intersecting crowd tubes yield a crowd. Circles represent transitions in human video textures.

Crowd tubes enable simple detection of collision and spatial separation constraint violations between a pair of crowd elements by performing distance thresholding on

temporally-corresponding circle proxies.Tighter-fitting proxy shapes, such as ellipses or bounded volume hierarchies, may also be employed for greater constraint violation detection accuracy. Crowd tubes are generated by translating and concatenating ground plane trajectories at transition points, thus forming a rigid path for 3D video billboards to follow. For each position in the path during animation, (a) the video billboard's object-centered coordinate system origin is updated to compensate for segmentation motion within the texture, and (b) the video billboard's origin is translated to its current 3D position (in the ground plane).

The artist authors a crowd by instancing crowd tubes which are anchored to specific points in time and space in the output video volume. A keyframe selected from a crowd tube's corresponding input clip V_i is displayed at the anchor point, thus providing for behavior control and a limited form of crowd choreography. Section 3.3.4 describes our interaction technique for anchoring and generating crowd tubes.

Extensions: In our experiments, we treated crowd tubes as rigid objects. However, crowd tubes can potentially be non-rigidly transformed and placed to achieve more variety in behavior. For example, stretching and squashing crowd tubes in the temporal direction correspond to slowing down and speeding up behavior playback. Behaviors may be recorded at high speed to decrease pedestrian speed without replicating or interpolating input frames. Furthermore, with additional analysis of segmented input video, crowd tubes could represent individual footsteps. This capability could support the accurate clustering of crowd tubes into supernodes to support the layout of entire lanes of pedestrian traffic or other aggregate crowd effects.

3.2.2 Unary Layout Constraints

The artist may specify polygons in the output view to represent a variety of layout constraints including traversable regions, entry-exit regions and obstacles. Polygons partition the output video volume into well-defined spaces which are used to accept

or reject crowd tube samples as a pre-process for satisfying unary constraints before binary constraint satisfaction. Figure 7 illustrates an example scene with two traversable polygons (drawn in green) corresponding to two sidewalks, and four entry-exit polygons (in red).

Given a crowd tube, which includes a concretely-positioned rigid trajectory in sidewalk coordinates, in-polygon testing of each point in the trajectory yields a binary trajectory mask indicating in-out status per point. The mask serves two purposes: (1) to decide whether a crowd tube legally enters and exits the scene, traverses the scene in a designed traversable region and does not intersect with an obstacle that is part of the background clip V, (2) to trim a crowd tube to only include trajectory points which lie inside a traversable region. The second purpose is important for efficient animation and rendering as extraneous concatenated segments of V_i would be needlessly animated and rendered.

A crowd tube must pass the following sequence of tests in order to be accepted before binary constraint satisfaction can occur. Testing is performed with trajectories and polygons mapped to sidewalk coordinates using the homographies described in Section 3.3.3. First, a crowd tube's trajectory mask is computed via in-polygon testing with all entry-exit polygons. Note that an entry-exit polygon can serve as a location for both scene entry and exit. For example, a pedestrian may enter from the left side of the screen and leave on the right or vice versa for a pedestrian moving in the opposite direction. If the mask has two connected components, it satisfies the entry-exit constraint. Next, the crowd tube is trimmed to traversable regions by selecting a range within the trajectory from the first in-traversable-region point to the last traversable point and removing points outside the range. The crowd tube is trimmed a second time to include the range of points starting from the first entry-exit point to the last. Finally, a crowd tube is deemed to satisfy unary constraints if no trajectory point lies outside a traversable polygon.

3.2.3 Behavior and Density Control

In the previous section, we described how to accept or reject a crowd tube sample subject to unary layout constraints. In this section, we introduce an interaction technique for generating a seed set of crowd tubes of artist-controlled behavior and density.

Behavior: The goal is to enable an artist to specify when and where specific keyframes of V_i should appear in the output video. In the spirit of video textures, we assume that infinite playback of V_i is possible by computing transition points within V_i and walking the motion graph comprised of a node per frame and a directed edge per transition. Chapter 5 describes the challenges and an in-studio approach to generating human video textures, or controllable animations made from joint video and motion capture of human motion. For the purpose of investigating the constrained layout challenges facing video-based crowd synthesis, we assume that each V_i has been pre-processed to identify transition frames.

Behavior control is made possible by specifying a crowd tube's anchor point in the output video volume. An anchor point is a 4-tuple (x, y, t_{out}, t_{in}) specifying that frame t_{in} of V_i should appear in the output video volume at (x, y, t_{out}). Given an anchor point, a crowd tube is generated by concatenating frames both forward and backward in time according to the motion graph associated with its clip V_i. In principle, multiple anchor points may be established to animate a crowd element using multiple keyframes. In practice, forwards-backwards concatenation is performed from a single anchor point using a random walk on the motion graph (or a simple loop). This simple keyframe-based approach to behavior control can enable a variety of tightly choreographed crowd outputs.

Density: How should an artist control the density of a video-based crowd? In model-based crowd synthesis, the crowd artist typically establishes an emitter location and flow rate in the style of particle systems. Agent-based simulation ensures that all

emitted crowd elements satisfy unary and binary layout constraints by virtue of the agent's dynamical model. In our video-based approach, the number and proximity of constraint satisfying crowd tubes are governed by the process of generating crowd tube candidates before constraint satisfaction, which we define as the *seed set*.

We loosely provide for density control by treating the seed set as an upper bound for the population and density of the crowd result. It is possible for all crowd tubes in a seed set to satisfy all constraints defined in this chapter, but this is unlikely to occur in space-time regions of high density. Therefore, the video-based crowd artist can approximate a desired density by generating a seed set whose density is proportional to the expected constraint-satisfying set. A high concentration of anchor points (and consequent crowd tubes) in a space-time region of the output will loosely define a desired density.

To support the otherwise tedious task of establishing an anchor point for each sample in the seed set, we created a "spray" interaction technique to distribute a set of crowd tubes with user-defined center and spread in the output video volume. The artist selects one of four bins of crowd tubes (1, 5, 50, or 100) to spray at a time. A specific V_i may be selected or randomly chosen for spraying. The keyframe t_{in} corresponding to V_i and spread in terms of standard deviation from t_{in} in frames are specified in the interface for normally distributing crowd tubes in the output video volume. A single click sprays the normally distributed crowd tubes with a spatial variance equal to the squared density (e.g. 25 for a spray of 5 tubes). Section 3.3.4 presents our interface for spraying.

3.2.4 Constraint Optimization Problem

After generating a seed set of crowd tubes, we must solve a constraint optimization problem: Given a seed set of crowd tubes, select a constraint-satisfying subset of tubes of maximal cardinality. The cardinality maximization objective is important

for controlling crowd population and density; as cardinality increases, the gap between the expected density and actual density decreases. As the artist increases the density of crowd tubes to occupy all free space in the output volume, the constraint optimization problem becomes a packing problem.

3.2.5 Conflict Graph and Independent Set

To solve this general constraint optimization problem, we observe that it can be reduced to a more specific computational task on a graph: the maximal independent set (MIS) problem. MIS is defined as the largest subset of nodes in a graph such that no pair share an edge. MIS is equivalent to computing the maximal clique in the graph's complement and is known to be NP-hard [71].

Given a set of crowd tubes, we construct a conflict graph G with a node per crowd tube and an edge per binary constraint violation. The seed set is pre-processed to instantiate a graph node for each crowd tube that satisfies unary constraints, as previously described in Section 3.2.2. An edge is inserted between each pair of nodes p and q if at least one of the following binary constraint violations occurs:

- A collision is detected at any point in time between p and q, which occurs if the L2 distance between p and q in sidewalk coordinates falls below a collision detection threshold t_c.
- If p and q are generated using the same input clip V_i (defining p and q as *clones*), the L2 distance between p and q in sidewalk coordinates falls below a spatial proximity threshold t_p.

Following conflict graph construction, we compute an approximate maximal independent set I using Luby's algorithm [71]. The solution set of crowd tubes I is then used to animate and render a video-based crowd. Animation and rendering details are presented in Section 3.3.5.

3.3 System

The crowd tube-based representation, constraint authoring interaction techniques, constraint satisfaction algorithm and rendering procedure were used to construct an interactive system for synthesizing video-based crowds. Figure 13 illustrates the linear pipeline. This section describes each of the steps in detail. The following section describes four experiments using our system and the chapter concludes with a discussion of our findings.

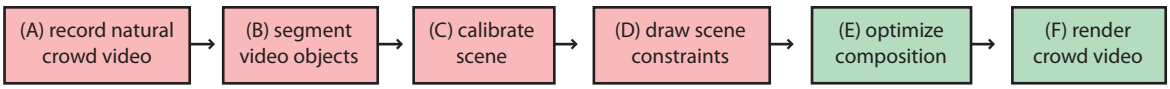


Figure 13: Pipeline for video-based crowd synthesis. Red boxes indicate steps requiring user interaction and green boxes are automatic.

3.3.1 Capture

To address the primary research question of how to layout segmented video objects while satisfying unary and binary constraints imposed by the 3D scene, we designed a video capture process to meet three objectives: (1) the crowd artist should be free to casually record video without having to direct actors to move in specific ways, (2) the crowd artist should not be required to insert specially constructed calibration objects into the scene and (3) the artist should be able to pan, tilt and zoom into the scene to capture an observed behavior with maximal resolution. The camera's motion is restricted to panning, tilting and zooming to support simple homography-based calibration, which is described in Sec. 3.3.3. The scene is also restricted to contain a single primary ground plane where synthetic crowds are to be added. In principle, our technique is extendable to multiple planes or, with a more complicated calibration process, to enable crowd tube trajectories to lie on the surface of arbitrary geometry. Crowd synthesis on rolling hills, for example, would require hill geometry and potentially more video clips to account for the greater variation in dynamics as

individuals climb up or speed down hills.

In our experiments, we captured four scenes using a tripod-mounted high definition video camcorder (specifically, the Sony VIXIA HFS100 model). Three of the scenes were recorded in an undirected manner for the purpose of populating the scene with natural behaviors. In the fourth scene, we directed an actress to validate the choreography process in a planned manner. Across all scenes, the capture process consisted of choosing a suitable vantage point, selectively zooming and locking in on desirable behaviors for synthesis, and choosing a background video. In principle, video may be captured while continuously panning, tilting and zooming. In practice, we chose to fix the camera's pose after adjusting the field of view separately for each clip V_i and for the background plate V. In future work, the camera's motion may be stabilized using automatically recovered homographies. Figures 19, 20, 21 and 22 illustrate example background views V and segmented behaviors from each V_i as time-lapse visualizations.

3.3.2 Segmentation

After recording the scene and its crowd elements, which include individuals, groups of individuals and inanimate objects, each crowd element's source clip V_i must be segmented for ground plane trajectory estimation and video billboard matting. Segmentation is one of the three key challenges facing video-based crowd synthesis. An accurate, temporally coherent result is necessary to create the illusion of a plausible synthetic crowd. Furthermore, a rapid, interaction-limited segmentation process is required to generate a variety of crowd tube clip sources V_i in a practical amount of time. There is a trade-off between artifacts associated with segmenting many clips of short duration and segmenting a lesser number of clips with longer duration. Shorter clips increase the frequency of transitions required to animate a crowd element moving from an entry to an exit (or staying in place for inanimate objects). On the

other hand, longer clips display transitions less frequently in the output but at the cost of lowering the quantity of behavior sources V_i, thus potentially damaging the perception of crowd variety.

Considering the quality versus quantity tradeoff, the ideal solution is a fully automated technique for segmenting V_i with high accuracy and temporal coherence. The crowd artist may capture multiple crowd elements in a single clip, V_i, when recording a zoomed in view of the scene with a particular target behavior in mind. Thus, our segmentation task requires specification of the target video object in the form of user interaction. Chapter 4 describes a segmentation technique which satisfies the requirement of minimal interaction; the user may specify the target in V_i by scribbling foreground and background areas on the first frame and the rest of the clip is automatically segmented. A more accurate result may be produced if the last frame is also labeled in this manner. The minimally-interactive technique presented in Chapter 4 does not satisfy our accuracy requirements, which are more strict in our special effects domain.

To achieve a greater level of accuracy at the expense of interaction time, the remaining results were segmented using a commercially available interactive video object segmentation system. Specifically, we used the "Rotobrush" tool in Adobe After Effects CS5. While the technique is unpublished, a number of interactive segmentation papers have been recently published by the Adobe company [11, 80]. We suspect the "Rotobrush" tool is based at least in part on technology described in their recent work, especially the impressive SnapCut system [11]. Table 2 reports the amount of interaction time required to segment several example clips in the *Embarcadero* scene, displayed in Fig. 20.

Table 2: Segmentation interaction time: This table reports time spent on interactive video stabilization and segmentation for several input clips in the Embarcadero scene. Note that segmentation time is not proportional to the clip duration, but rather a function of the image content. Challenges to segmentation include foreground-background color overlap and large shape changes.

sequence	image	frames	stabilization	segmentation
bluesuitman	Fig 20b	233	0 min.	47 min.
womanleft	Fig 20d	240	8	245
groupof4	Fig 20e	340	7	215
roundtable	Fig 20f	77	5	24

3.3.3 Calibration

Calibration is necessary to animate video billboards on a 3D ground plane in the scene. To summarize, calibration enables the following three step process for constructing and animating crowd tubes. First, ground tracks extracted from each segmented clip V_i are mapped to sidewalk coordinates. Second, a crowd tube and its entry-to-exit trajectory is generated by concatenating segments of the ground track in the sidewalk frame. Finally, crowd tube sidewalk trajectories are mapped to the output view and raycasting produces a 3D entry-to-exit trajectory planted in the ground plane. Figure 14 illustrates how the three coordinate frames are related by homography through the ground plane. The three coordinate frames are exploited during the crowd tube construction process, illustrated in Figure 15.

Our interactive process consists of two main steps: (1) estimation of ground plane parameters A, B, C, D corresponding to the $Ax + By + Cz + D = 0$ representation and (2) estimation of a series of homographies (see Fig. 14): output-sidewalk homography $\mathbf{H_{os}}$ (1 per scene) and input-output homography $\mathbf{H_{io}}$ (1 per clip V_i). During animation, a homogeneous 2D point $\mathbf{x_i}$ in the zoomed-in input frame V_i is mapped to a point $\mathbf{x_o} = \mathbf{H_{io}x_i}$ in the output view V. Given a virtual camera placed at the 3D origin with principal point $\mathbf{x_c}$ (set to the center of V) and focal length f (set arbitrarily), a 3D ray is constructed with a direction towards output point $\mathbf{x_o}$ and

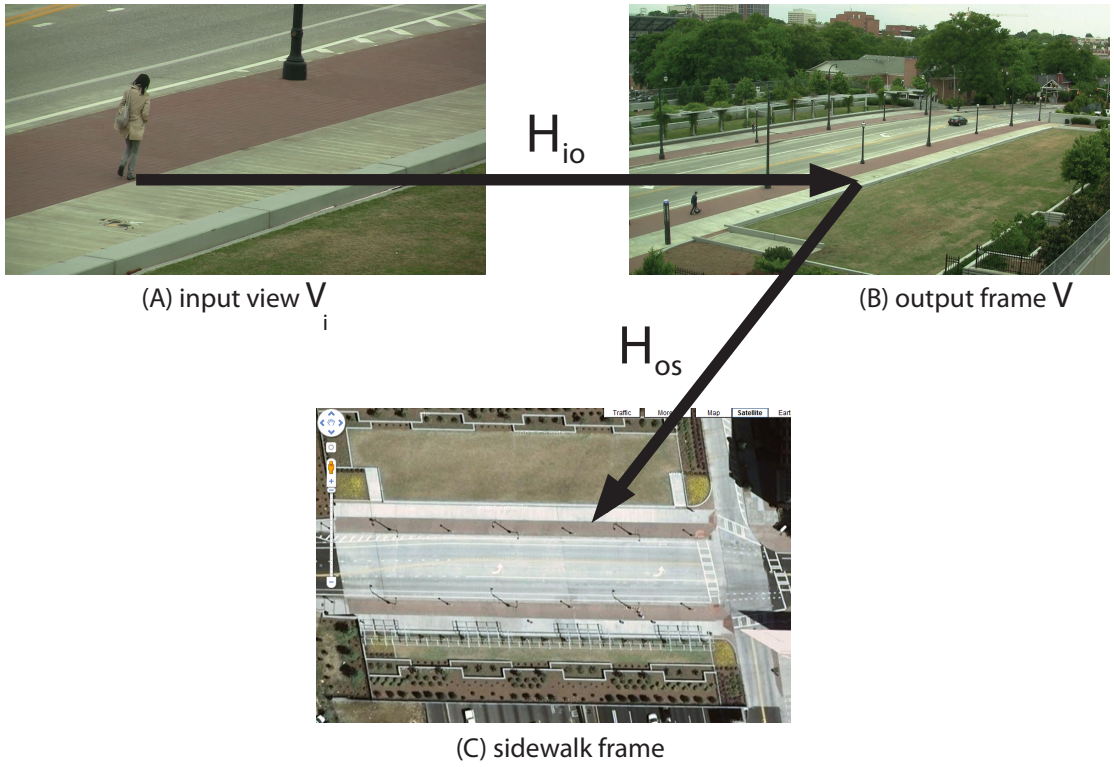


Figure 14: Homographies mapping 3 views. (A) Camera is zoomed to view V_i to capture behavior at high resolution, (B) the extracted ground track is mapped to output view V under $\mathbf{H_{io}}$ and (C) mapped to sidewalk view under $\mathbf{H_{os}}$.

cast through the calibrated ground plane to retrieve the billboard's 3D position $\mathbf{X}$ per crowd tube and per output frame.

Ground plane: We simplify the interactive ground plane calibration process by making a reasonable assumption about the scene: at least four objects of approximate fixed height, such as lamp posts or adult humans, are standing orthogonal to the ground plane. The object r that is furthest away is chosen as a *reference* whose base point is fixed at $z = f$ and the remaining objects' 3D base points are estimated relative to r. Based upon this assumption, we solve for $\mathbf{x} = [A, B, C, D]^T$ parameters in a least squares manner: Find $\mathbf{x}$ that minimizes $\|A\mathbf{x}\|$ where

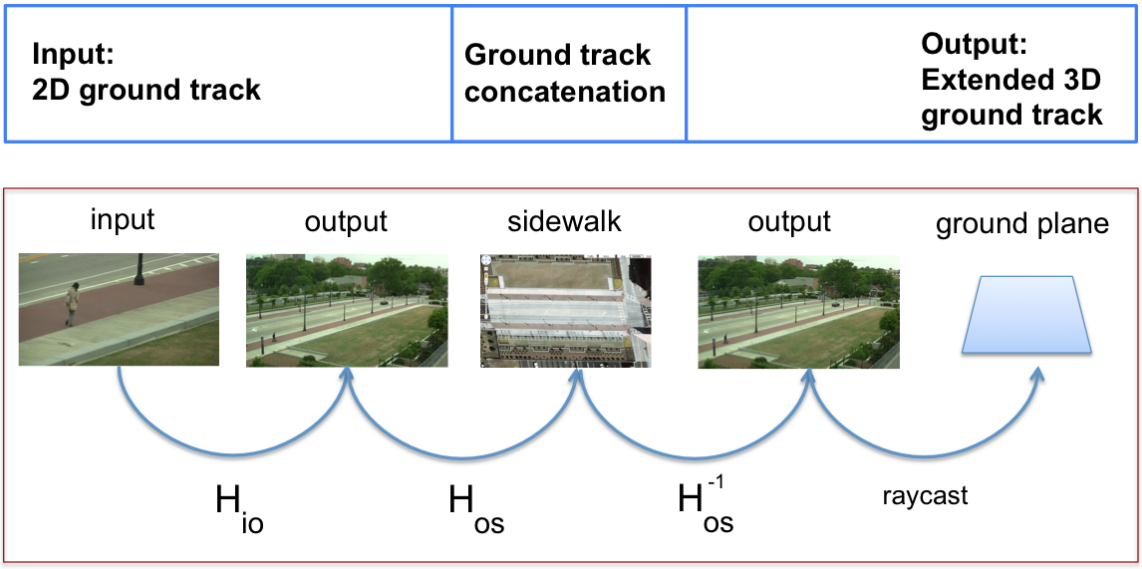


Figure 15: Trajectory concatenation to generate a crowd tube. Given an input video clip with a segmented behavior (woman on left wearing beige in this example), the 2D ground track is extracted automatically and mapped to sidewalk coordinates using the homographies illustrated in Fig. 14. Sidewalk trajectories are concatenated to extend the track from scene entry to exit. Following concatenation, the extended track is mapped to 3D ground plane coordinates by ray-casting the trajectory's output coordinates through a calibrated ground plane.

$$A = \begin{bmatrix} s^1 x_b^1 & s^1 y_b^1 & s^1 f & 1 \\ s^i x_b^i & s^i y_b^i & s^i f & 1 \\ \vdots & \vdots & \vdots & 1 \\ s^n x_b^n & s^n y_b^n & s^n f & 1 \end{bmatrix}$$

and where $s^i = \|x_b^r - x_t^r\| / \|x_b^i - x_t^i\|$ is the ratio of the reference object r's height to non-reference object i's height in the output frame V. User interaction in the form of at least eight mouse clicks on V provides feature coordinates corresponding to the bottom point $[x_b^i \ y_b^i]^T$ and top point $[x_t^i \ y_t^i]^T$ for four or more objects in principal point centered coordinates $\mathbf{x} - \mathbf{x_c}$. At least three or more non-reference objects' 3D base points are determined relative to r using the fixed height and orthogonality assumptions. The objects should be distributed to span the area where synthetic crowds are desired, as opposed to concentrated in one small region of the area, to

avoid errors caused by over-fitting.

Homographies: The user supplies four point correspondences between each V_i and V to estimate $\mathbf{H_{io}}$ using Direct Linear Transformation [41]. An additional set of four point correspondences are manually specified to estimate $\mathbf{H_{os}}$ for mapping output view V to sidewalk frame V_s. As shown in Fig. 14, we obtain top-down views V_s of each scene using publicly available aerial images. There are several techniques for estimating metric-rectified homographies [41] if an indoor scene or outdoor scene lacking aerial imagery is desired for crowd synthesis. We chose a simpler approach as most metric rectification techniques require additional information, such as five or more orthogonal line pairs planted in the plane.

3.3.4 Crowd Authoring

After capturing, segmenting and calibrating a scene, the artist designs a video-based crowd by annotating the output view V with a set of polygons followed by crowd tube instantiation. We constructed a crowd authoring interface (Fig. 16) to enable annotation and to give the user a basic pre-visualization of their crowd. A typical crowd authoring session involves the following procedure. First, the user draws polygons representing traversable regions where crowds may lie. Likewise, entry and exit zones are demarcated with polygons. The green polygon in Fig. 16 represents an example traversable region and the red boxes indicate entries and exits, whose meaning depends on a crowd tube's direction of travel. Polygons serve as unary constraints which act on rigid trajectory samples, as previously described in Sec.3.2.2. Second, a crowd is constructed by clicking in V to *spray* crowd tubes into the scene, thus enabling a limited form of behavior and density control over a crowd composition. Finally, the system solves the constraint satisfaction problem (Sec. 3.2.4) and animation and rendering follows.

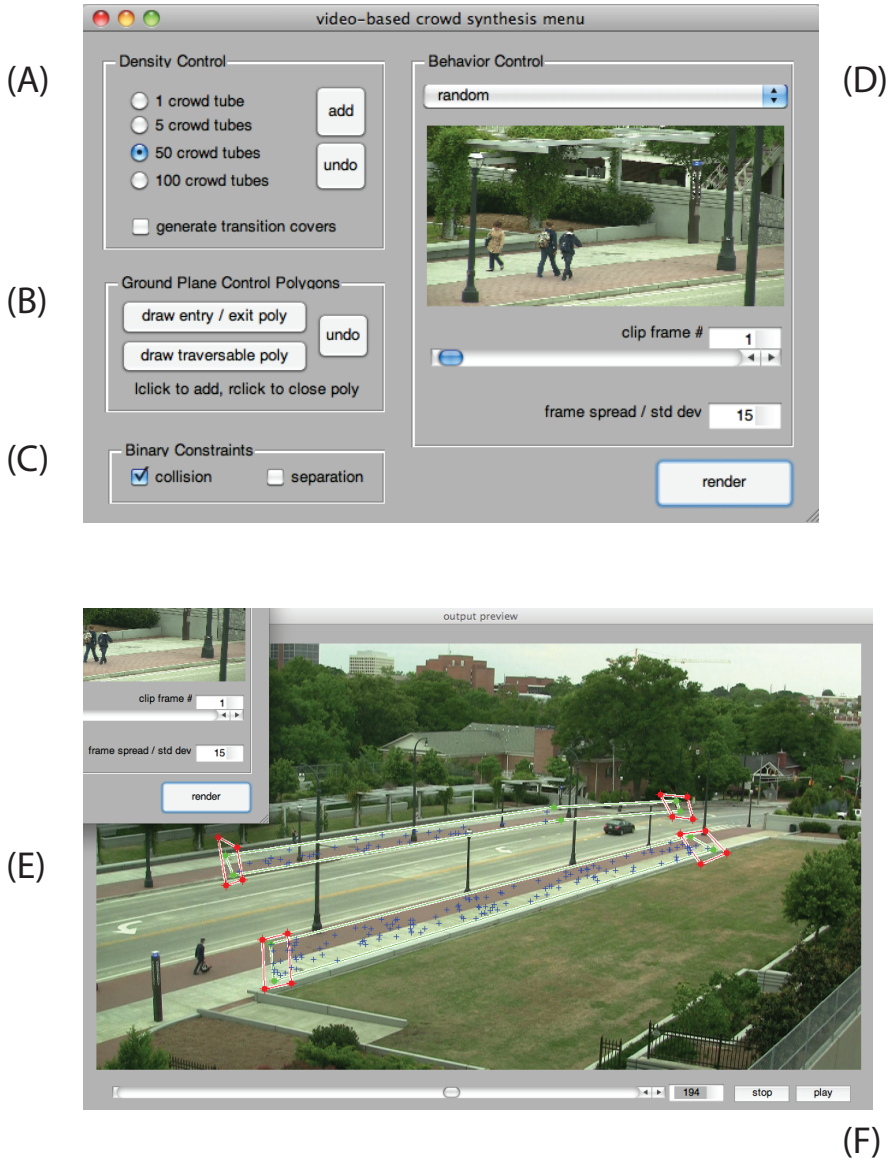


Figure 16: Crowd authoring interface. (A) Four density levels are available for distributing crowd tubes in the output as a batch, (B) Polygons represent entry-exit regions and traversable regions for constrained crowd tube placement, (C) Collision avoidance and spatial separation of clones are two binary constraints which can be activated before synthesizing a video-based crowd, (D) A user may choose from a set of segmented input clips V_i for instantiating crowd tubes, (E) Unary constraint polygons and crowd tube positions per frame may be previewed in the output window, (F) An output view's time slider allows for crowd tube placement at specific points in time and crowd pre-visualization, where each crowd tube's position is marked with a blue cross.

3.3.5 Animation and Rendering

In order to support the computationally expensive process of animating and texture mapping high definition video billboards against a video background plate, we built our system on top of an industry-standard video composition system. Following crowd authorship and constraint optimization, our system generates a script which is then processed by Adobe After Effects CS5 for automatic animation and rendering of output video. Before script execution, the user must set up an environment in the commercial application with a list of video assets and corresponding matte frames associated with each input source V_i. Script execution and rendering is a batch, offline process which takes between one and four hours in our experiments. The artist may exploit several features provided by the commercial system to improve the look of the crowd. For example, we produced virtual shadows for the *Bridge* scene by rotating, blurring, scaling and alpha attenuating segmentation clips. Figure 17 shows a screenshot of the rendering and animation engine.

3.4 Dataset and Results

Using our system, we produced a series of video-based crowds to demonstrate the capabilities of our crowd tube representation, constraint satisfaction algorithm and synthesis pipeline on four outdoor scenes. Our primary experimental objective was to answer the following research question: Given a video of a sparsely crowded scene, how can we synthesize a plausible video of the same scene populated with crowds of artist-controlled density and duration?

Introductory discussion: The main qualifier of this research question is the word *plausible*, and its meaning varies depending on the intended use of the proposed system. A successful approach should at least produce a result that appears plausible when used in the most forgiving context; at a glance and from afar, does the crowd look real? At what observation duration does plausibility break down? What are the

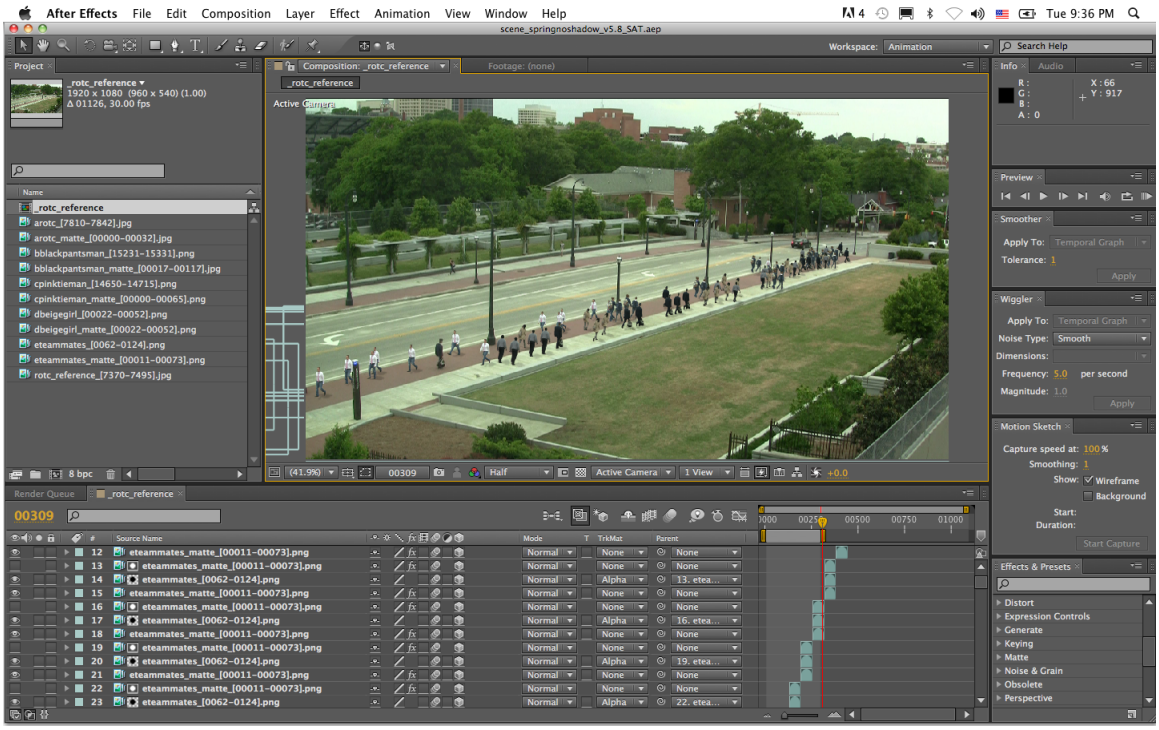


Figure 17: Commercial video composition software screenshot. Adobe After Effects CS5 serves as the rendering and animation engine in our prototype system. The upper left displays a list corresponding to V_i, the input video clips. Layers are clearly concatenated as displayed by the staggered structure of layers in the bottom. The video billboard composition is visible as a preview image in the center.

artifacts that cause the crowd to look fake? To investigate these subjective issues, we generated a range of video outputs and collected a set of observations which we provide in this section.

Bridge: Figure 19 depicts a still image result from the *Bridge* scene and corresponding dataset, which is comprised of five video clips V_i, a crowd-free background clip V and calibrated ground plane and homographies. A still image can reveal basic aspects of compositional quality including perspective correctness, segmentation accuracy, traversability satisfaction, lighting and shading, and crowd variety. Figure 19 reveals a crowd of two women in beige, two men in a bright shirt, two pairs of friends walking together, three men carrying a backpack and three men in grey sweatshirts. The difference in height between the man in the beige sweatshirt on the left and

the same man far on the right reveal a correct perspective effect. No pieces of the body appear to be missing but the poses of the two men appear to match. However, the pose of the same man in the middle is different. None of the crowd elements appear to collide, or occupy the same space in the ground plane. Clones are spatially separated. Figure 18 reveals that binary constraints are being successfully enforced by our algorithm. These issues demonstrate that basic compositional objectives are being met.

They also hint at how our video-based crowd synthesis system may be used to control a large number of synthesis variables with a tight level of control. Thus, the system lends itself to extensive perceptual studies in the spirit of recent perceptual studies of factors which affect crowd realism [73, 74]. In contrast with those works, which employ non-photorealistic model-based avatars, our system presents the opportunity to control and investigate more subtle properties of photo-realism and video-realism for the first time.

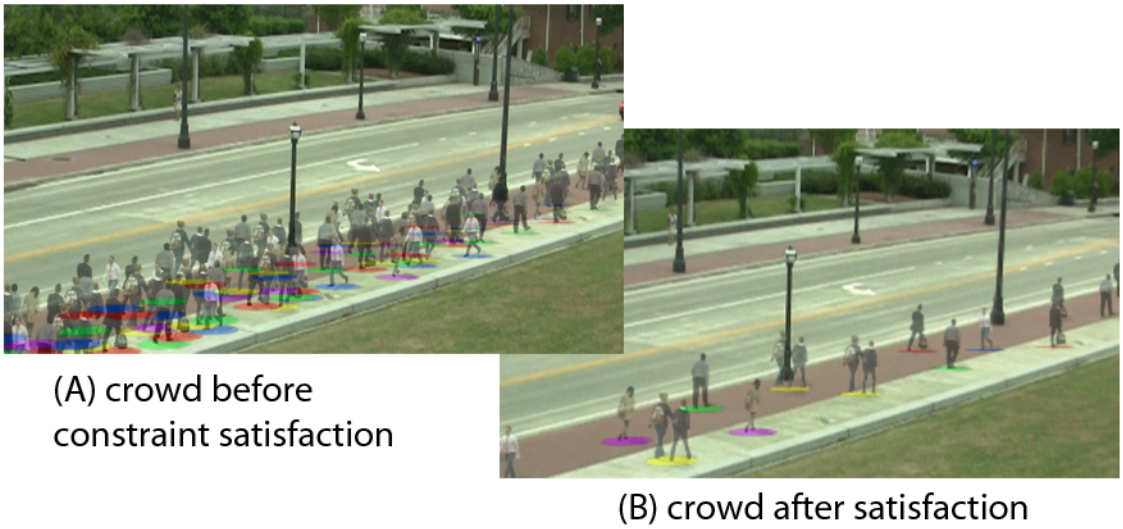


Figure 18: Crowd visualization before and after constraint satisfaction. (A) All crowd tubes are visible, showing an impossible real-world composition of pedestrians in motion. Circle proxies are color-coded by clip. (B) A constraint-satisfying set of crowd tubes: no crowd element collides and clones are spatially separated along the ground plane.

Table 3: Crowd tube sample counts before and after constraint satisfaction: The authoring interface was used to quickly generate a random crowd in 40s by spraying batches of 100 random crowd tubes per mouse click. Of 361 unary constraint-satisfying crowd tubes, Luby's algorithm computed an approximate MIS of cardinality 61 in 50.44s. Animation and rendering took approximately 1hr 40mins for a 10s result.

sequence	**sample count**	**sample count in MIS**
beigepants	97	23
cameraman	77	8
bluesuitman	101	25
groupof4	8	0
textingwoman	1	0
womanleft	74	5
roundtable	3	0
total	361	61

Embarcadero: We designed a second dataset, titled *Embarcadero*, to observe the effect of increasing the quantity, duration and variety of crowd elements. In contrast with *Bridge*, a major pathway for students crossing campus, *Embarcadero* is a popular tourist destination in San Francisco, thus presenting more opportunities to capture casual behaviors of visitors. Figure 20 presents a still from a *Bridge* result and corresponding dataset, which consists of seven clips spanning a range of dynamics. Three of the input clips (Fig. 19a,b,d) contain linear walking trajectories. Fig. 19g shows a tourist holding a camcorder while standing still, followed by walking away after recording. The remaining clips include static crowd elements including: (e) a group of four visitors who have stopped so that one man can place his sweater in another's backpack, (f) two women having a discussion at a roundtable and (c) a woman texting while seated on the ground. Table 3 provides crowd tube counts and crowd authorship, animation and rendering timings.

Sailboats: A third scene titled *Sailboats* was selected to investigate the issue of crowd output quality versus authorship time. While *Embarcadero* depicts a wide

variety of complex behaviors at the cost of approximately one full day of interactive segmentation work and rendering time, *Sailboats* includes only two light-colored sailboats moving in opposite directions against a blue background (Fig. 21) placed at a distance. A crowd of two sailboats was constructed from capture to render time in less than 3 hours of work. Segmentation required little to no interaction, likely due to little shape change and foreground-background overlap.

Park: Finally, we captured and processed a fourth scene (see *Park* illustrated in Fig. 22) as an example of planned crowd choreography. Inspired by typical crowd effects involving battling armies, we aimed to produce a "battle" of umbrella-wielding individuals who approach each other as a marching army and stop short of clashing to open an umbrella. We varied the shape of the line of both moving fronts at umbrella-opening time to demonstrate our system's behavior control capabilities. Umbrella-opening keyframes were instantiated with a standard deviation of 15 frames (using interface controls displayed in Fig. 16d and f) into the output video volume in a designed fashion. Figure 22 displays the result for the *Park* scene. Video results demonstrate our system's ability to choreograph a behavior-controlled crowd of this nature.

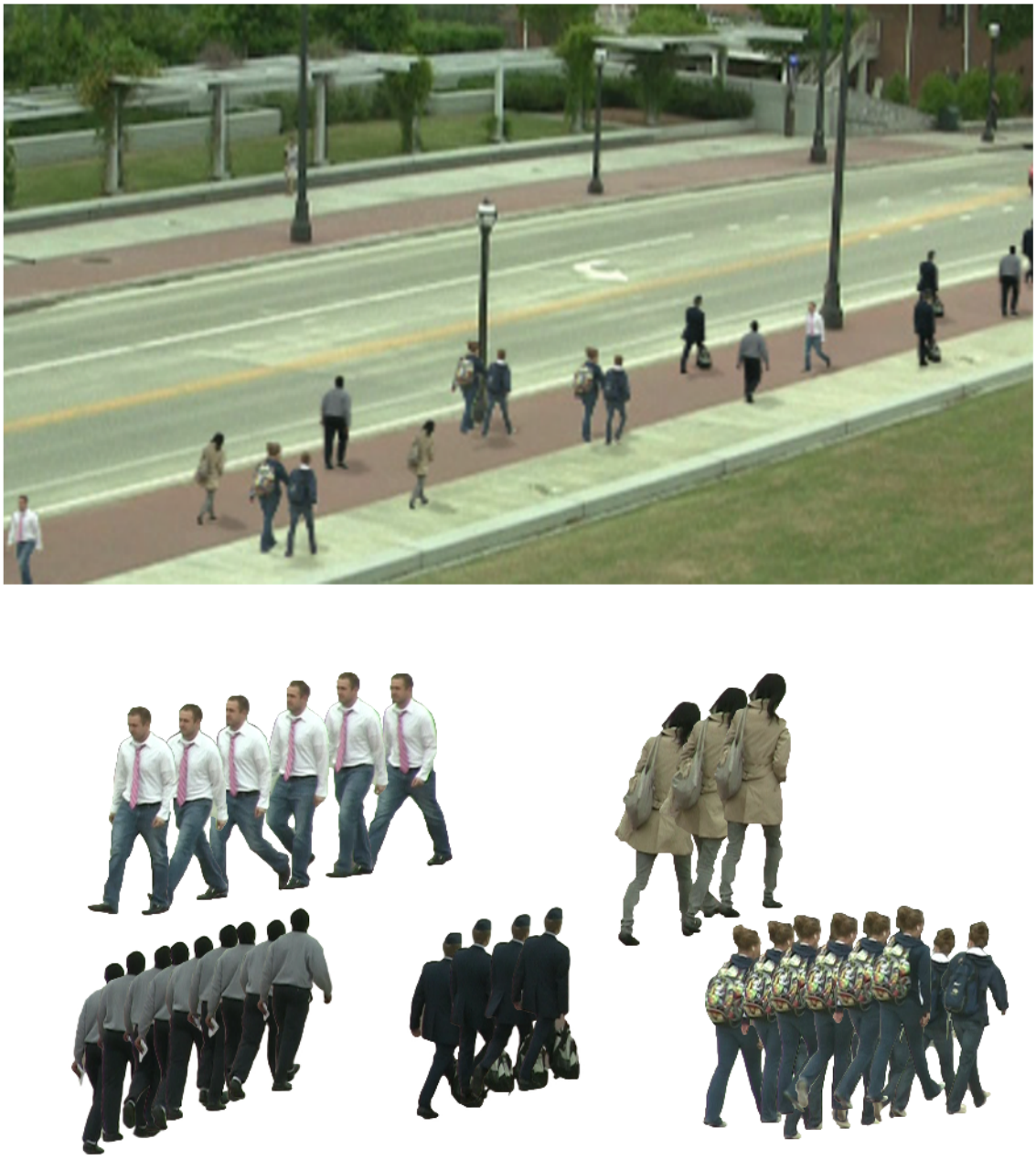


Figure 19: Bridge result and dataset

Figure 20: Embarcadero result and dataset: Interactive segmentation timings for clips B, D, E and F are provided in Table 2.

Figure 21: Sailboat result and dataset

Figure 22: Park result and dataset

3.5 Qualitative User Evaluation

In addition to generating a range of crowds across four scenes to demonstrate the capabilities of our prototype system, we designed and conducted a crowd authoring exercise to assess the system's usability, bottlenecks and favored features. The primary goal was to qualitatively investigate how effective the system is in allowing someone to realize their "vision" of a video-based crowd animation with little to no prior experience with the system. Three users who had previous experience with advanced video editing software were asked to plan, author and evaluate a video-based crowd using the *Embarcadero* dataset.

Design Rationale: By constraining the users to create crowds within 30 minutes and with a limited palette of seven behaviors for a single scene, we hoped to discover positive and negative trends across all users' experiences. Specifically, we wanted to uncover common planning and design strategies for coping with the same limited set of input examples. Would two or more users construct a similar crowd design? If so, would the similar designs be driven by the same rationale? After planning their crowds, would the users encounter the same bottlenecks in executing their plans? If so, would they identify workarounds or suggest improvements to overcome interface problems? To quickly explore these questions, we designed the following set of steps as a qualitative pilot study. Our intent was to identify and address issues with the crowd authoring interface to inform and motivate specific research directions in future work.

3.5.1 Steps

Each user executed the following steps:

1. The artist spent fifteen minutes in an orientation session to understand how to use the traversable control polygon tool, behavior control tool, crowd tube insertion tools and how to preview their output. All behaviors were previewed

and the keyframe-based crowd tube insertion mechanism was demonstrated to show how an input keyframe (Fig. 16d) appears at the clicked spatial position in the output keyframe (Fig. 16f).

2. After the user demonstrated correct understanding of input and output keyframes, she was oriented to frame spread control (Fig. 16d, at bottom) which perturbs the input keyframe by with a controllable offset. After the user understood how frame spread can be used to rapidly insert crowd tubes with out-of-step motion, she was oriented to the batch crowd tube insertion tool (Fig. 16a).

3. Following orientation, the user was left alone for ten minutes to plan a crowd animation by drawing the scene and the intended layout on a sheet of paper. We instructed the user to plan a crowd which they thought could be created in less than thirty minutes of interaction with the authoring interface. The user was also given freedom to create a crowd of their choosing; they were free to choreograph either a realistic (could naturally occur) or unrealistic (dramatic or surreal) composition.

4. After sketching a crowd plan, the user was instructed to think aloud and ask questions while using the interface to execute their plan.

5. Following authorship, the users took a 30-45 minute break while their crowds were rendered. They were then asked to comment as to whether their crowd met their expectations.

3.5.2 Results

Crowd sketches for each user are provided in Figures 23, 25 and 27. Figures 24, 26, and 28 show select output frames and their captions describe the verdicts.

Summary of Verdicts: Users 1 and 3 were satisfied with the result while User 2 was not. User 2 had a similar goal to User 1; both setup a cluster of static obstacles

in the center to see how the moving pedestrians would interact with them. User 1 created a larger collection of static obstacles in the center than User 2 and as a result, more static obstacles survived constraint satisfaction for User 1. User 3 designed and satisfactorily authored a crowd of men in suits walking toward a single female to see what it looked like when "she disappeared in the crowd." The remaining three subsections include comments from each user.

User 1: The following quote summarizes User 1's experience of authoring crowds with limited behaviors: "I had seen some of the crowds generated before, and was aware of the roughly 6 character limitation, so I had decided on trying to play-up these limitations and attempt to make a more weird geometric design with my crowd. In my mind I had pictured that the scene would appear as strikingly unnatural with the stationary sprites forming odd shapes in the center while hordes of clones marched into the center. What came out was even better than I had imagined. At first glimpse it really seemed as if nothing was out of the ordinary, but then further study revealed the actually bizarre nature of it. I really enjoyed this effect as it seemed to sublimate my original, garishly overt design into a more subtle surreal portrait. I also think we discovered something about clones in that, in general, neighboring clones tend to break the illusion, but when many clones are horded together and their walk-cycles are simply offset they appear quite natural."

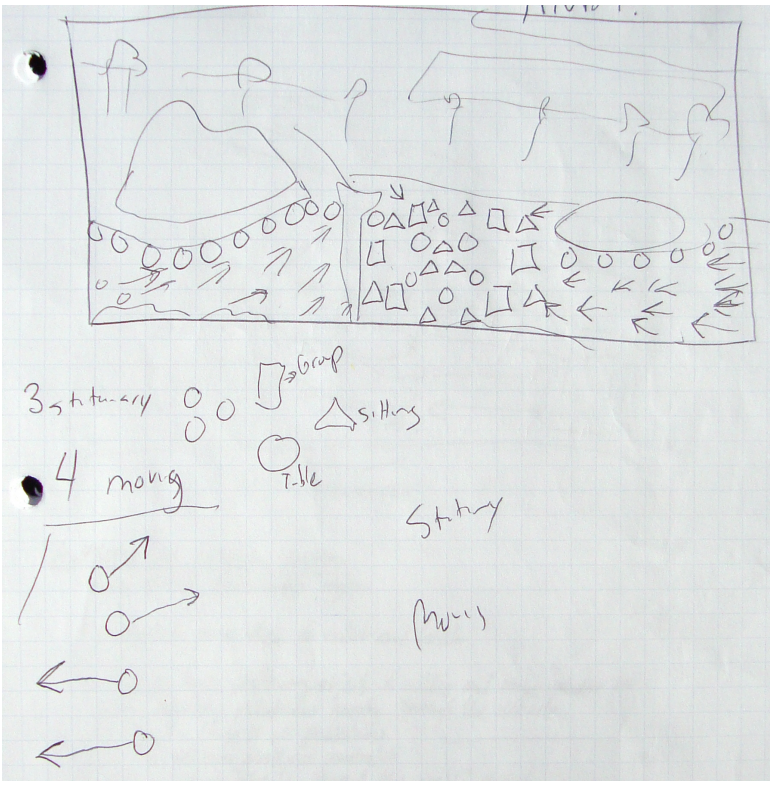


Figure 23: Crowd plan for User 1: The user intended to have a large collection of static objects in the center while hordes of pedestrians marched from both sides towards the center while slipping through the obstacles upon arrival.

Figure 24: Select output frame for User 1: The output indicates that the general goal described in Figure 23's caption was met; a large section of immobile behaviors are exhibited in the center while groups march in from the sides.

User 2: Similar to User 1's goal, User 2 planned a crowd which would exhibit pedestrians marching through a line of static obstacles. Constraint satisfaction removed all but three static obstacles positioned near the center of the frame. The result did not meet the expectations in the corresponding sketch exhibited in Figure 25. User 2 stated that after viewing the result, he wished he could have forced many of the static obstacles to remain in the constraint-satisfied solution set. He also mentioned that if he could force inclusion of specific crowd tubes, it would help if the interface highlighted those which must stay in the output in a different color from those which may vanish following constraint satisfaction.

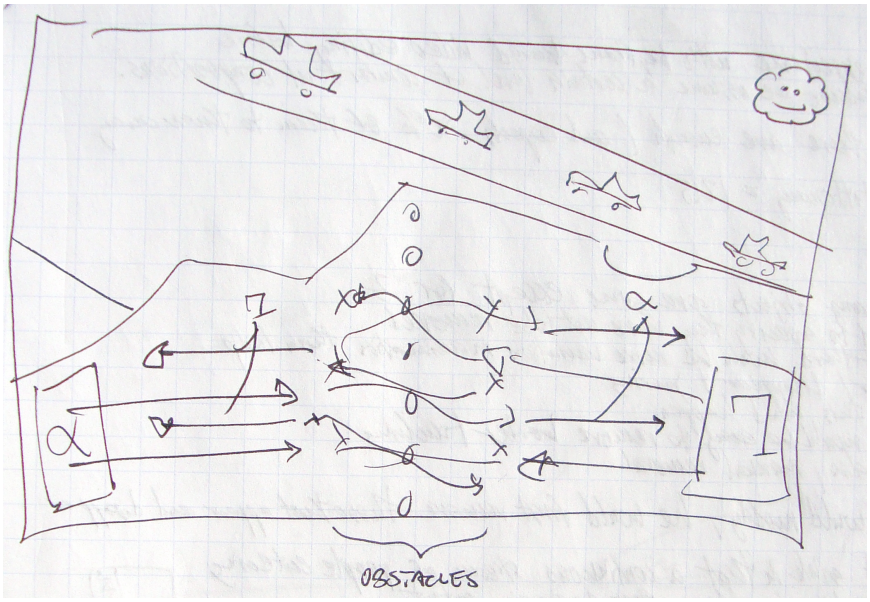


Figure 25: Crowd plan for User 2: Similar to User 1's goal, User 2 planned a crowd which would exhibit pedestrians marching through a line of static obstacles.

Figure 26: Select output frame from User 2: Constraint satisfaction removed all but three static obstacles positioned near the center of the frame. The result did not meet the expectations in the corresponding sketch exhibited in Figure 25.

User 3: When User 3 was asked if she had achieved her goal, she responded: "Yes, but I didn't understand that there would be others [pedestrians] in the background. I thought the tools were clear to use and I liked spraying the area. The only difficulty I had was not knowing the direction they would exactly walk in. I had a general idea of which directions of movement I needed to meet my goal, but I was limited to only a few which did not walk directly towards each other."

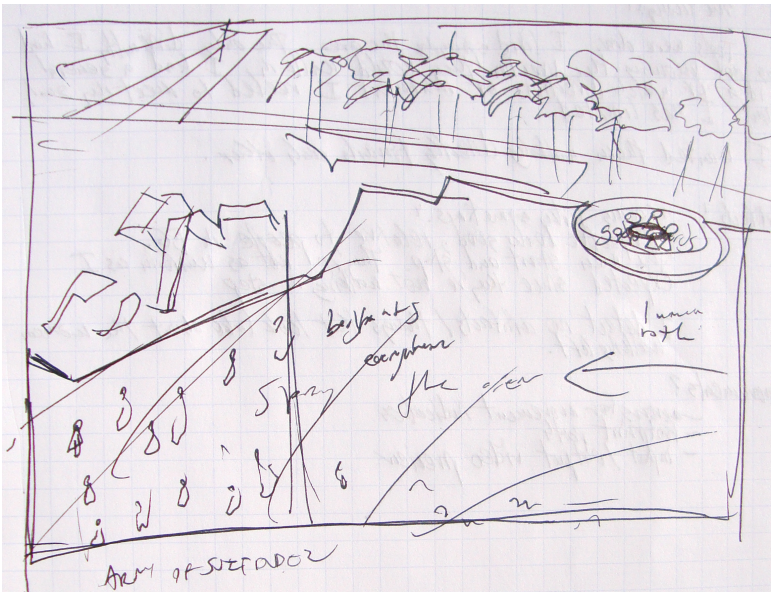


Figure 27: Crowd plan for User 3: The user's intent was "to embrace the surreal and create a large group of suit-wearing men walking to a woman moving in an opposing direction to see if the woman disappeared among them."

Figure 28: Select output frame for User 3: The result met User 3's expectations.

3.5.3 User Trends and Current Limitations

In this section, we highlight several common problems and desired features among all users who evaluated the system. We extracted the problems and feature ideas

from think aloud comments and questions asked during the authorship and evaluation steps. Some of the trends include specific observations for new features and improvements to the process and interface for authoring a video-based crowd animation.

- **Crowd tube layout confusion:** All three users asked clarifying questions regarding how frames would appear in the output for frames other than the desired output keyframe. Upon understanding that the remaining spatial positions of a crowd element are pre-defined by the captured behavior, two of the users suggested adding "footprint" icons indicating the full trajectory of a crowd tube as the mouse is moved inside the traversable region.

- **Behavior inspection and organization:** All three users planned their crowd by first reviewing the available behaviors and sorting them by degree of motion (static vs. moving along ground). Also, all three users asked to re-preview each behavioral clip to understand how the element will move and made note of the motion on a sheet of paper. They also all suggested adding arrow icons below each clip preview (Fig. 16d, at top) to view a summary of the clip's motion without having to watch it.

- **Iterative editing and constraint satisfaction:** Early on in the authoring process, all three users expressed a desire to preview their crowd output after constraint satisfaction had occurred. The prototype system does not currently support iterative adding, solving and previewing the crowd in progress and this proved to be the most important limitation for the users. The users knew that their final output would only include a subset of the inserted behaviors. Thus, all three resorted to casually composing their crowd with a greater focus on positioning and movement of blocks of crowd elements than the less-achievable goal of meeting an exact output population target.

- **Behavior removal:** Another related issue was a common desire to remove individuals and groups of individuals from the composition. Crowd tube removal was not possible with the interface. Upon learning this, one of the authors started over and worked more carefully to insert crowd tubes.

- **Sampling from behavioral categories:** The users also expressed a behavior control-related issue concerning the random addition of crowd tubes. Two of the users noted that "random picking with keyframe control is a problem because you want specific keyframes per behavior" and "[it] would be nice to categorize behaviors according to moving and not moving and then randomly selecting within those groups." This suggests that users would like a behavior-organizing process which would allow them to control batch insertions of crowd tubes in a more semantically meaningful manner.

- **Surreal crowd design:** Finally, all three users felt that the limited number of behaviors in the *Embarcadero* dataset forced them to create surreal and dramatic crowds. They wanted to either access or capture and process additional behaviors to achieve their plan which was originally conducted with full knowledge of the limited palette of behaviors available for authoring.

In summary, the user evaluation revealed several limitations related to behavior preview, control and general crowd authorship with a small number of behaviors. Based on our findings, we believe there are two crucial areas of improvement for future work on the crowd authoring processing and interface: (a) behavior categorization and visualization features would partially address the primary issues raised concerning crowd planning and behavior selection and (b) having the human in the loop with iterative constraint satisfaction would ease concerns regarding the difference between planned and constraint-satisfied crowds.

3.6 Limitations

In addition to the specific crowd authoring interface issues uncovered in the previous section, there are several limitations to general video-based crowd synthesis. First and foremost, the variety of crowd outputs is limited to behaviors observed at capture time. This is a limitation facing image-based rendering techniques in general. While increasing the size of the dataset would alleviate the problem, it comes at the expense of interactive segmentation and calibration time.

Second, transitions artifacts appear in the form of popping artifacts as our current system uses basic jump cuts between frames out of sequence. Transition synthesis techniques presented in Chapter 5 address the problem but they currently require an elaborate motion capture setup against a blue screen backdrop. We attempted to address the transition visibility problem in early experiments by searching for independent sets which explicitly minimize transitions, but this resulted in distracting crowd tube layout patterns. The most common pattern was a coupling of video billboards which roughly travel in lockstep.

Third, crowds are animated atop a single ground plane in our prototype system, thus eliminating a large category of interesting scenes.

Fourth and finally, footskate artifacts become exacerbated as video billboards are positioned further away from their originally captured location. This is caused by deviations in the angle between the camera-object line at capture time and the object's line of travel direction when recorded. For example, a subject walking in the depth direction along the camera's principal axis at capture time would exhibit footskate when animated to appear in the corner of the image. As crowd density increases, the issue becomes less problematic however.

3.7 Summary of the Chapter

This chapter described how to synthesize video-based crowds by animating a collection of matted video billboards. We presented a constraint satisfaction problem and solution for placing video billboards in a 3D scene subject to constraints on motion and appearance. We designed a system and algorithm for composing a constraint-satisfying set of *crowd tubes*, our representation of segmented videos coupled with a trajectory of ground occupancy proxy shapes (e.g. circles) planted in a ground plane. The presented results include the first demonstration of video-based crowd synthesis and timings for an interactive synthesis pipeline. These results portray three natural, un-choreographed scenes and one planned and directed crowd. In addition to providing a range of crowd outputs from four scenes, a qualitative user evaluation of the prototype system uncovered issues and directions for future work on the crowd authoring process and interface.

CHAPTER IV

VIDEO OBJECT SEGMENTATION USING TEMPORALLY-COHERENT LOCAL GRAPHCUTS

The previous chapter described and addressed the constrained layout challenges facing video-based crowd synthesis at the video object level of detail. The layout problem was approached under the assumptions that (a) crowd video may be accurately segmented with good temporal coherence and (b) segmented video clips may be reanimated in a visually smooth manner without noticeable transitions. To experimentally validate our approach to video billboard layout and animation, we used a commercially available video object segmentation tool provided as part of Adobe After Effects CS5. Following semi-automatic segmentation, we manually identified transition points and employed basic jump cuts to transition between concatenated segmented video clips.

This chapter presents a novel method for interactively segmenting crowd tubes using temporally-coherent graphcuts and a unique ground-truth database for evaluating segmentation-based tracking algorithms. Automatic video object segmentation is fundamentally important to scaling up our video-based approach to crowd synthesis. Crowd variety comes at a cost of segmentation interaction time and this chapter makes important steps toward the goal of video-based crowd synthesis using large databases of crowd behavior.

4.1 Introduction

Recent work in visual target tracking has explored the interplay between state estimation and target segmentation [16, 28, 81]. In the case of active contour trackers and

level set methods, for example, the state model of an evolving contour corresponds to a segmentation of target pixels in each frame. One key distinction, however, between tracking and video object segmentation is that a tracking system must operate automatically once the user has manually specified a target of interest. In contrast, systems for video object segmentation [11] are designed to be interactive and incorporate guidance from the user during the analysis process.

Motivated by applications in surveillance, several recent works have achieved excellent results for on-line tracking in real-time [16, 28]. However, the quality of the segmentations produced by on-line trackers are in general not competitive with those produced by systems for interactive segmentation [11, 80, 69], even in cases where the user intervention is limited. One factor is the use of global optimization methods, such as graphcut, which can obtain the optimal segmentation within a single frame. In contrast, tracking methods evolve a solution over time from a given starting point, do not have optimality guarantees, and are subject to the accumulation of error.

This chapter describes a target tracking method based on the sequential application of graphcuts to the video volume. Our goal is to obtain higher-quality segmentations than previous on-line methods, without requiring significant user interaction. We are motivated by applications such as biotracking, where an off-line batch formulation of video analysis is acceptable, but guidance by the user must be minimized in order to be acceptable to biologists. In such applications, it is highly-desirable to be able to reliably segment the limbs of a target animal, in order to support analysis of behavior, but it is not necessary to obtain the pixel-accurate segmentations that are needed in video post-production and special effects domains.

On-line tracking methods use a variety of techniques to achieve robust performance, including adaptive cue combination [16], spatially-varying appearance models [28], and shape priors [26]. In contrast, our goal is to leverage the single cue which is the most representative of tracking problems, namely motion coherence, within

a volume segmentation framework. The two main elements in our approach are an incremental forwards-backwards graphcut formulation and a technique for dynamically adapting the temporal graph structure via motion estimation. Our approach is straight-forward and easy to implement, and does not require significant parameter tuning. In many cases it can yield higher segmentation quality than existing on-line trackers. However this improvement comes at the cost of greater computational requirements within a batch formulation.

A second goal of this work is to support quantitative assessment of segmentation quality in tracking through the development of a standardized database of videos with ground-truth segmentations. There has been no systematic quantitative or comparative evaluation of segmentation quality within the visual tracking literature.[1] We identify three properties of video sequences that can hamper segmentation: color overlap between target and background appearance, interframe motion, and change in target shape. We have developed a quantitative measure for each of these properties, and have assembled an evaluation dataset, called *SegTrack*, which spans the space defined by these challenges. We provide quantitative and qualitative evaluation of our method and compare it to two recent on-line contour-based trackers [16, 28]. We will make our SegTrack dataset and our tracking code available from our project website.

In summary, this chapter makes three contributions:

- We introduce an incremental graphcut-based formulation of video tracking which provides high-quality target segmentations and can handle extended video sequences.

- We study the instantiation of temporal n-links within the graphcut framework

[1] In contrast, there has been extensive work on comparing the performance of standard state-based trackers. Some representative examples are [51] and the VS-PETS workshops.

and demonstrate that dynamic motion-based instantiation gives superior results.

- We present a novel database that supports systematic quantification of segmentation quality with respect to three types of challenges found in real-world video footage.

The chapter is organized as follows. After presenting related work in Sec. 2, we describe our tracking algorithm in Sec. 3. Next, we present and discuss the SegTrack database in Sec. 4 followed by quantitative and qualitative experiments in Sec. 5. Finally, we discuss limitations in Sec. 6 and conclude in Sec. 7.

4.2 Related Work

There are two bodies of previous work which are related to our method. The first are techniques for video object segmentation and layer segmentation which also make use of a graphcut formulation. The second are tracking methods which make use of the segmentation of the target object.

The graphcut segmentation method has been applied to volumetric datasets [21, 64], including video [22]. In the formulation from [22], the MRF is instantiated in the temporal dimension by linking identical pixel sites in adjacent frames (see Fig. 31(B)). This approach was improved by Li et. al. [69], by creating links between superpixels in adjacent frames, as illustrated in Fig. 31(D). In comparison, we demonstrate that instantiating temporal links on the basis of motion estimation gives better performance for many challenging video sequences. There are a few works utilizing KLT-based optical flow in the context of segmentation. In these works, flow estimates are used to compensate for motion[48], or modify the cost functions in a fixed graph structure [80], while in our case we use KLT tracked feature points to dynamically-instantiate the graph topology in the temporal dimension.

Several recent works in interactive segmentation of video objects have demonstrated excellent segmentation quality. A representative example is the SnapCut system from Bai et. al. [11]. A well-designed interface coupled with an effective optimization framework supports local refinement by the user and the propagation of corrections across the sequence. There are two characteristics of our approach that distinguish it from these previous works. First, we present a simple and effective approach to enforcing temporal coherence via constraints from partial solutions within a series of volumetric cuts. In contrast, the more common approach is to apply graphcut to single-frame cost maps, and achieve coherence and adaptation via the manipulation of the cost values. While this approach has a great deal of flexibility and is therefore suitable for an interactive system, it requires a large number of parameters and leads to a complex algorithm. Second, we investigate multiple strategies for instantiating temporal n-links and show that an adaptive technique gives the best performance.

A large number of on-line tracking methods can produce object segmentations (representative examples are [28, 16, 99]). Since these methods were designed for on-line, real-time performance, they do not address the same goal as our off-line batch analysis method. However, these methods are designed for fully-automatic operation, and therefore represent an interesting point of comparison. Bibby and Reid describe an impressive tracking system in [16], which demonstrates adaptation of the target model and integration of multiple cues so as to track a wide range of challenging targets. A level-set based system, described in [28], uses a combination of spatially-constrained appearance modeling and motion estimation to achieve good segmentation performance. In comparison to these works, we leverage a batch formulation of the problem along with volumetric cuts to obtain higher quality segmentations at the cost of additional computation. In addition, we conduct the first quantitative and qualitative comparisons between these existing methods using a standardized testing

set with ground-truth. Other works which address active contour models and optimality [89], and more advanced optimization methods [55], are interesting topics for future study.

4.3 Algorithm

Given a video sequence and manual segmentations of a target of interest in the first and last frames, our goal is to carve the moving target out of the video volume, yielding a target segmentation at every frame. We adopt the standard volumetric graph cut formulation, in which the volume is represented as a Markov Random Field with hidden nodes corresponding to the unknown labels $\{\text{fg}, \text{bg}\}$, of foreground and background pixels respectively. Given accurate foreground and background appearance models, it is possible to segment a video object through a set of independent graph cuts, one for each frame. However, this approach will not work well in challenging sequences, where the appearance of the target changes significantly over time. Thus a key question in video object segmentation is how to enforce the temporal continuity of the target motion.

In the original volumetric graphcut formulation of Boykov and Jolly [22], additional temporal n-links are added for pairs of corresponding pixel sites in temporally-adjacent frames, $(p_t, p_{t+1} \in T)$. The resulting optimization problem is to find the label assignment f to p that minimizes

$$\begin{aligned} E(f) &= \sum_{p \in P} D_p(f_p) \\ &+ \lambda_N \sum_{(p,q) \in N} V_{pq}(f_p, f_q) \\ &+ \lambda_T \sum_{(p_t, q_r) \in T} V_{p_t q_r}(f_{p_t}, f_{q_r}) \end{aligned} \tag{1}$$

where (p_t, q_r) denotes a temporal n-link between site p in frame t and site q in frame r, and λ_N and λ_T are user defined weights. In this section, we describe how to apply volumetric graph cuts and how to instantiate the temporal n-links T for best tracking

performance.

4.3.1 Forwards-Backwards Sweeps of Graphcuts

Standard graph structures used to minimize Equation 1 over all candidate segmentations in a video volume are constructed to represent the entire video, as opposed to a segment of video. Therefore, a mincut on the video volume results in a segmentation of all video frames. In our formulation, we define a *local* graph that only represents a sub-volume corresponding to a sliding window of frames in time.

After user initialization, our method executes a forwards and backwards series of graphcuts on the local volume (see Fig.29). Before proceeding, a foreground and background GMM is estimated from the union of the initial segmentations using EM. Each of the forwards-backwards sweeps begin with the window centered on their respective manually segmented frames. Hard constraints are established for the central frame of the window using labels obtained from the previous volumetric cut (or from the interaction in the beginning). Soft constraints are then entered throughout the volumetric graph in the form of t-links and n-links. T-link costs are entered in the standard way [21] as a function of foreground (background) color likelihood given by the GMMs. Standard 8-connected spatial n-links are entered for all nodes in the local graph using the cost function given in Equation 2

Temporal n-links are entered as a function of motion estimation as described in the next subsection. After entering all soft and hard constraints into the volume, a mincut assigns labels to all pixels in the volume and the sliding window proceeds to the next set of frames. Each frame in the video is visited up to M times where M is the size of the window along the temporal axis, which we set to 5 in our experiments. The forwards-backwards sweeps produce two video object segmentations (one per sweep). The intersection of these segmentations are output as the track result.

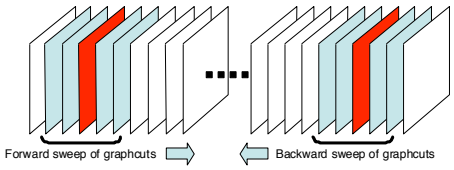


Figure 29: Forwards-backwards sweeps of local graphcuts. The red frame corresponds to the central frame in a local graph specified on the sub-volume indicated in blue. Hard t-link constraints are established on the red frame and soft t and n-links are entered for nodes on the blue frames.

4.3.2 Temporal N-Links from Feature Tracks

A standard feature detector [91] and tracker [19] is used to identify and instantiate temporal n-links between the central frame and all other frames in the local volume as illustrated in Fig. 31E. Basic outlier rejection is performed by thresholding out tracks longer than a multiple (we use 5) of the mean length. We connect a set of n-links from a patch (5x5) centered on feature p in frame t to its corresponding patch centered on feature q in frame t'. Costs associated with temporal n-links are specified by equation 2 in the same fashion as the standard spatial n-link costs. T-links for the central frame are entered as hard constraints using the labels obtained from the previous volume segmentation. After entering all soft and hard constraints into the local volume, a mincut produces a segmentation over the volume and the sliding window proceeds to the next set of frames. This has the effect of propagating high quality segmentation information from the initial manually specified segmentation to the other end of the entire video volume.

$$V_{p_t q_r}(f_{p_t}, f_{q_r}) = \begin{cases} exp(-\frac{\|I_{p_t} - I_{q_r}\|^2}{2\sigma^2}) & f_{p_t} \neq f_{q_r} \\ 0 & f_{p_t} = f_{q_r} \end{cases} \quad (2)$$

4.4 SegTrack Database

In addition to developing a simple and effective graphcut-based tracker, the second goal of this work is to facilitate a deeper understanding of the tradeoffs and issues

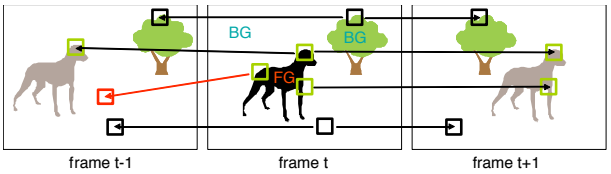


Figure 30: Temporal coherence from tracked features. Nodes (pixels) in the central (shown as red in Fig. 29) frame are connected to the other frames' nodes according to KLT flow. Temporal n-links are entered from a patch of pixels at frame t to its corresponding patch at $t^{'}$.

involved in on-line and off-line formulations of video segmentation and tracking, via a standardized database of videos with ground-truth segmentations. There has been very little comparative work addressing the segmentation performance of tracking methods. Our starting point was to identify three properties of video sequences that pose challenges for segmentation quality: color overlap between target and background appearance, interframe motion, and change in target shape. We developed quantitative measures of these phenomena and carefully assembled an evaluation dataset, called SegTrack, which spans this space of challenges. We also provide direct comparison between our batch tracking method and two state-of-the-art on-line contour-based algorithms [16, 28].

In order to obtain a set of sequences which adequately cover the space of challenge metrics, we went through the following selection procedure. First, a set of 11 image sequences were manually screened and selected to fulfill desired challenge combinations. Challenge ratings for each selection were then manually predicted to be either high or low and the selections were assigned into one of eight combination bins (2 extremes per challenge for 3 challenges). Next, the image sequences were manually segmented and measured by computing each challenge metric as described below. Finally, six total image sequences ranging in length from 21 to 70 frames were selected to maximally cover the challenge space (see Table 4). We justify and quantify our metrics below.

Target-background color overlap: An accurate segmentation of the target,

provided by the user, is commonly used to estimate a color model for the target and non-target pixels. Unfortunately, the discriminative power of such models are inversely proportional to the degree of overlap between the figure-ground color distributions. Numerous trackers and interactive segmentation systems evaluate color overlap to decide how and when to lessen the importance of color and increase reliance on other models of the target (e.g. a locally modeled shape prior as in [11]). We chose to model target and ground colors with GMMs containing 5 Gaussians. Equation 3 gives a formula for evaluating color overlap on a per-frame basis. High C_1 values correspond to large target-background overlap, which makes segmentation and tracking more difficult. The average measure per sequence is given in Table 4.

$$C_1 = \frac{\int_{X \in tg} p(X|bg)}{\int_{X \in tg} p(X|tg)} + \frac{\int_{X \in bg} p(X|tg)}{\int_{X \in bg} p(X|bg)} \tag{3}$$

Interframe target motion: Both global and local models of target and non-target motion are commonly estimated in an attempt to precisely localize color, shape and other cues for target segmentation and state estimation. When motion estimation proves difficult, e.g. in areas of homogenous texture, large target motions can potentially corrupt local appearance models. From ground truth segmentation, we measure interframe motion as the foreground XOR intersection area normalized by the mean object bounding box area. The per-frame average motion is reported in Table 4.

Target shape change: Shape priors constructed from target initialization, keyframes (as automatically obtained in [106]) and recently segmented frames are often adaptively applied when other appearance models (e.g. color) are predicted to have small discriminative power. When target shape varies little and motion estimation is reliable, shape priors have been demonstrated to track through areas of large figure-ground color overlap and occlusions [11]. However, when motion estimation is unreliable or shape change is drastic, this strategy can fail for obvious reasons. We provide

Table 4: SegTrack database metrics and scores: These sequences correspond to the following difficulty levels for color overlap, interframe motion and shape change: (a) *parachute*: low-low-low, (b) *girl*: low-low-high, (c) *monkeydog*: low-high-high, (d) *penguin*: high-low-low, (e) *bird*: high-high-low, and (f) *cheetah*: high-high-high. Scores correspond to average number of error pixels per frame. Scores A and B are taken from the bi and uni-directional pass versions of our method, resp. A uni-directional pass score represents a single sweep of our algorithm initialized on one end only. Select frames from *parachute*, *girl*, *bird* and *cheetah* are illustrated in Fig. 34 while frames from *penguin* and *monkeydog* are displayed in Fig. 32.

sequence	**color**	**motion**	**shape**	**A score**	**B score**	**[28] score**
parachute	.038	.119	.024	212	236	502
girl	.205	.145	.147	1097	1072	1755
monkeydog	.299	.243	.132	310	299	683
penguin	1.02	.016	.013	555	665	6627
bird	.466	.283	.070	262	270	454
cheetah	.760	.273	.187	669	641	1217

such challenge combination scenarios in the SegTrack database. Shape change is estimated in a similar manner to our motion metric; we quantify it as foreground XOR intersection area normalized by the mean object bounding box area after compensating for translational motion estimated from centroid differences. Table 1 reports the mean shape change for each sequence.

4.4.1 Discussion

In addition to selecting and measuring sequences intended to stress test 3 challenging aspects of segmentation-based tracking, we carefully tuned and benchmarked the competing level set-based tracker of [28] alongside two versions of our method. Our system is suited to offline batch processing while the system of [28] is an online single-frame initialized method. In the interest of maximizing fairness and completeness in quantifying the performance differences between these systems, we chose to report the output from a uni-directional pass, referred to as score B, using the same target segmentation used to initialize the tracker of [28]. Score A was produced using both forwards and backwards sweeps as described in Sec. 3. Both scores A and B were able to outperform that of [28] in terms of per-pixel segmentation accuracy. The large

differences in score for *penguin* and *cheetah* were caused by tracker failure, in which case the target contour either remained in place (*cheetah*) or vanished completely (*penguin*). We expect other trackers to follow this convention in case of tracker failure in reporting SegTrack benchmarks.

4.5 Experiments

In this section, we provide additional evidence of the benefits of our approach. We first quantify the effect of varying temporal n-link structures within our framework of sweeps of local graphcuts. Second, we give qualitative examples of our system's ability to generate more accurate segmentations than [16] and [28]. Finally, we demonstrate our system's capability to track long sequences and examples with occlusion.

4.5.1 Temporal Link Instantiation

We investigated the effects of varying temporal n-link structures by designing and testing five experimental conditions all within our framework of forwards-backwards sweeps of local graph cuts. Of these five experimental conditions (see Figure 31), two are novel (C,E) and three are not (A,B [22],D [69]).

The standard use of temporal n-links consists of connecting nodes that represent pixels in frame t to nodes of the same spatial location in frames $t-1$ and $t+1$. We classify links between nodes in the same spatial position in the temporal direction as *static* temporal n-links. Conversely, we classify links between nodes that are not necessarily in the same spatial position as *dynamic* temporal n-links. Dynamic temporal n-links are estimated as a function of the image content.

Li et al. [69] presented another example use of dynamic temporal n-links in their graphcut framework. In their work, the image sequence is oversegmented into a set of super-pixels. Each graph node represents a super-pixel and temporal n-links are entered from each node in frame t to all other nodes in $t-1$ and $t+1$ whose centroids lie within a Euclidean distance threshold (such as 15). Their method is the basis for

experimental condition D, which is depicted in Figure 31D.

The first novel condition (C) we tested consists of extending static temporal n-links beyond the frames immediately before and after a given frame to all frames in a five frame window. Table 5 reports the effect of condition C on video object segmentation. Relative to previously reported n-link structures (A,B,D), condition C results in better segmentations for all but one of the tested videos (*monkey*).

The second novel condition (E) was constructed to investigate the merits of feature tracking on large motion sequences in the context of temporally-coherent local graph cuts. Feature tracking follows a sparse set of detected features over many frames. Therefore, it is well-suited as a source of dynamic temporal n-links which span more than one frame in time. We used a standard feature detector[91] and tracker[19] to instantiate these n-links. Results show that condition E performed best across all sequences except the *monkey* video.

Condition D resulted in the worst performance across the treatments we tested. When a super-pixel is assigned the wrong label, the error increases by the area of the super-pixel. Without the interaction features designed to accompany this n-link structure to correctly label large error sites, condition D does not lend itself to achieving high per-pixel segmentation accuracy.

In comparison to the other sequences, results for the *monkey* video was not consistent with our findings. This sequence exhibits a challenging reflection in the water which thwarts motion estimation techniques for obvious reasons. Ground truth segmentations did not include the reflection regions by design.

4.5.2 Qualitative Comparison

Figure 32 illustrates differences in segmentation quality across a sample of sequences from video results provided from [16] (see Fig. 32A-C) and [28] (32D,E). To compare our output with [16], we selected all visible frames in the author's provided video

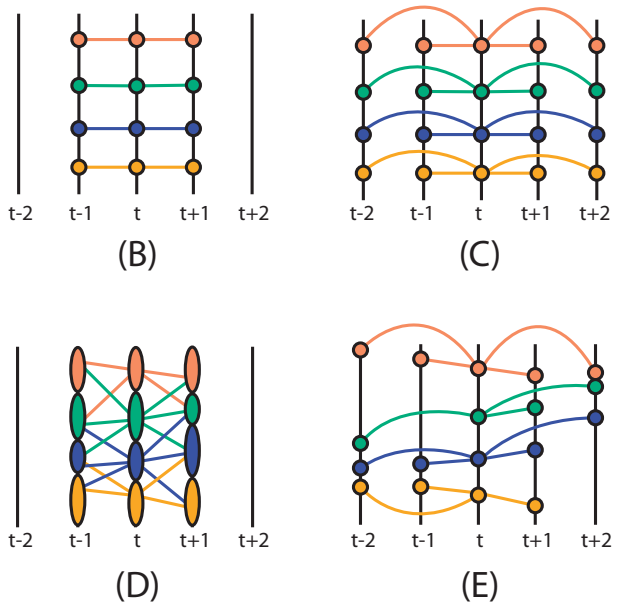


Figure 31: Temporal n-link structures: (A) no temporal n-links (not shown) (B) static temporal n-links in a 3 frame window, (C) static temporal n-links in a 5 frame window, (D) dynamic temporal n-links on super-pixel nodes as described in [69] and (E) dynamic temporal n-links instantiated from tracked features. Colored circles and arcs represent corresponding graph nodes connected by temporal n-links. Ellipses represent super-pixel nodes[33]. Static n-links connect nodes with the same spatial location. Dynamic n-links connect nodes as a function of the image content.

and initialized the bi-directional version of our tracker by labeling the first and last frames. Our method maintained track of the target while providing more accurate segmentations. Qualitative comparison to [28] was more straight-forward since the author provided all test sequences on the web, removing the need to identify the same sequences from BBC's Planet Earth.

In Figure 34, we show a greater variety of example tracker outputs. The upper four sequences are a selection of frames from our SegTrack database while the remaining two bottom clips were long sequences from BBC's Planet Earth. Full length outputs are provided in the supplementary video.

Table 5: Quantitative results: This table reports average number of error pixels per frame in four video sequences under five experimental treatments. Each treatment corresponds to the temporal n-link structure depicted in Figure 31. Treatment A corresponds to a graph without temporal n-links. The minimum error is highlighted in bold. Note that condition E results in minimum error for all sequences but *monkey*.

sequence	A	B	C	D	E
cheetah	999	1116	891	1459	**874**
girl	3539	2919	2793	3364	**2217**
soldier	3021	1576	1394	4841	**1375**
monkey	5580	**2435**	3384	8726	3459

Figure 33: Examples with occlusion

4.5.3 Occlusions

We have tested our system on several sequences exhibiting partial and complete occlusions. As suspected, we found that our method can handle them when color overlap is low and when occlusions are short in duration. See Fig. 33 for illustrations of such cases.

4.5.4 Segmenting Multiple Video Objects

We tested our method's ability to segment multiple video objects undergoing occlusion by segmenting one object first and another object second (see top of Fig. 34). Since our method exhibits good per-pixel accuracy, it affords the ability to exclude

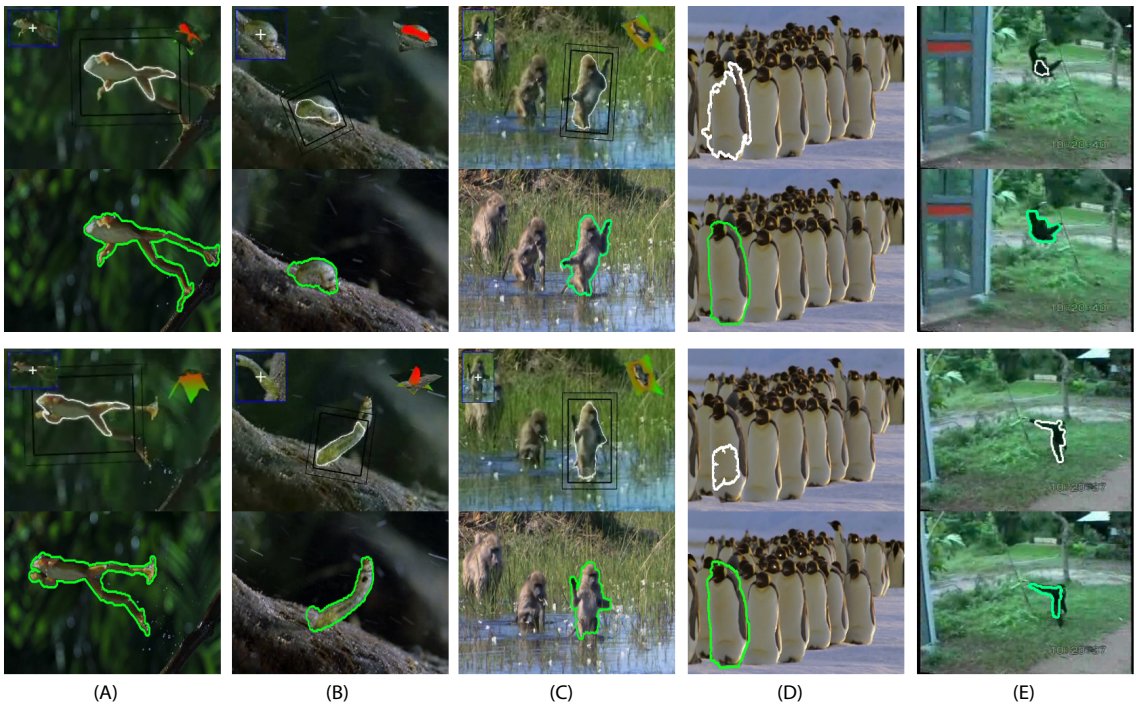


Figure 32: Comparative results: Each pair of images show a competing tracking method(above, in white) along with the proposed method (below, in green) for representative frames. Columns A-C: Comparison of output from [16] (rows 1 and 3) with ours (rows 2 and 4). Columns D-E: Comparison of [28] with ours (rows 2 and 4). Our method is more capable of producing accurate segmentations in general, but this is not always the case as illustrated in rows 3-4 of (E).

pixels in subsequent target segmentations that would normally act as a source of noise in motion and color estimation. More specifically, previously segmented video objects are: (a) used to establish hard constraints that force their pixels to be labeled "background", (b) excluded from both foreground and background color model estimates and (c) used to reject any feature tracks going into our out of previously segmented objects. During occlusions, especially when only a few pixels are still visible, it is important to mask out the occluder as best as possible. Our method successfully segmented through the challenging occlusion case exhibited in frames 7-11 in the *cheetah* sequence.

4.6 Limitations

While our system can achieve tracking results with high segmentation quality on a variety of datasets, there are a few limitations. First, we have observed some trailing and leading foreground label "fronts" which carry the effect of expanding the contour outside the true target boundary. This is caused by KLT outliers which leap from foreground to background and vice versa, allowing hard constraints to propagate onto and off of the target. Second, tracking quality is sensitive to small deviations in initialization, creating an interaction task that may last more than a minute before tracking may begin.

4.7 Summary of the chapter

We have described an off-line method for target tracking through sequential segmentations of the video volume. We leverage the good performance of the graphcut segmentation algorithm in an incremental formulation, thereby bounding the memory requirements of the method and making it suitable for video analysis. We present a ground-truth dataset for target tracking, called SegTrack, which is based on a systematic assessment of the sources of difficulty in accurate segmentation. We compare our method to two recent on-line trackers, and demonstrate improved performance. Our results suggest that, for off-line applications of tracking, it is possible to obtain more accurate segmentations at the cost of increased computation. We also study the question of how to best instantiate temporal n-links within the graphcut framework, and propose dynamic strategy based on motion estimation. Our code and the SegTrack dataset is available from the project website [98].

Figure 34: Qualitative tracking results: Top: *Cheetah* sequence exhibits difficult foreground-background color distribution overlap, occlusions, large target deformations and large camera motion. Our method can successfully segment both the predator and prey. Row 2: Bird sequence from SegTrack, exhibiting color overlap, large motion and small shape change, Row 3: *Girl* sequence[1] from the UCF action database, illustrating shape changes followed by Parachute, the easiest sequence in SegTrack. Rows 6 and 7: Two successfully tracked long sequences.

CHAPTER V

HUMAN VIDEO TEXTURES

The previous chapter addressed the challenge of segmenting captured video clips with good temporal coherence. Temporal coherence is of fundamental importance to video-based crowd synthesis as incoherence may include and exclude pedestrian parts over time in a visually noticeable manner and damage synthetic crowd plausibility.

In this chapter, we describe the problem of transition identification and synthesis for the controlled re-animation of segmented video clips. In the spirit of Chapter 4, we study the human video textures problem outside the context of crowds and at the individual level. Rather, this chapter focuses on the problem of concatenating clips which exhibit complicated articulations and non-rigidity while maintaining a visually smooth appearance.

5.1 Introduction

While both motion capture and video data can capture aspects of realistic human movement, current techniques for manipulating this data fall short of the goal of creating photorealistic controllable humans. Motion capture data encodes the realistic dynamics of human movement and can be used to synthesize realistic animations, but the task of endowing the resulting 3-D characters with a photo-realistic appearance is still quite challenging. Likewise, video data implicitly captures the complex relationships between movement, lighting, and appearance, but existing techniques for synthesizing novel video sequences from captured video have not been successfully applied to complex human movement. Techniques based on stereo or multi-view geometry have been used to provide interactive camera control during the replay of captured movement sequences [107, 31, 100], but these techniques do not address the

problem of synthesizing novel movement sequences from a corpus of examples.

A promising approach to generating photo-realistic video output is to rearrange the frames in a captured video sequence, thereby providing a limited ability to generate controlled animations [88, 87, 7, 27]. Frame rearrangement depends upon the ability to select good transition points and interpolate video frames across these transitions without introducing visual artifacts. Unfortunately, standard image-based similarity metrics, such as squared pixel differences, fail to choose good transition points when applied to video clips of human motion. These metrics do not capture the complex changes in appearance that are characteristic of humans in motion, especially with related articulations and non-rigidity. Furthermore, secondary motions like hair tousling and the crumpling of clothing add to the complexity of measurement of similarity.

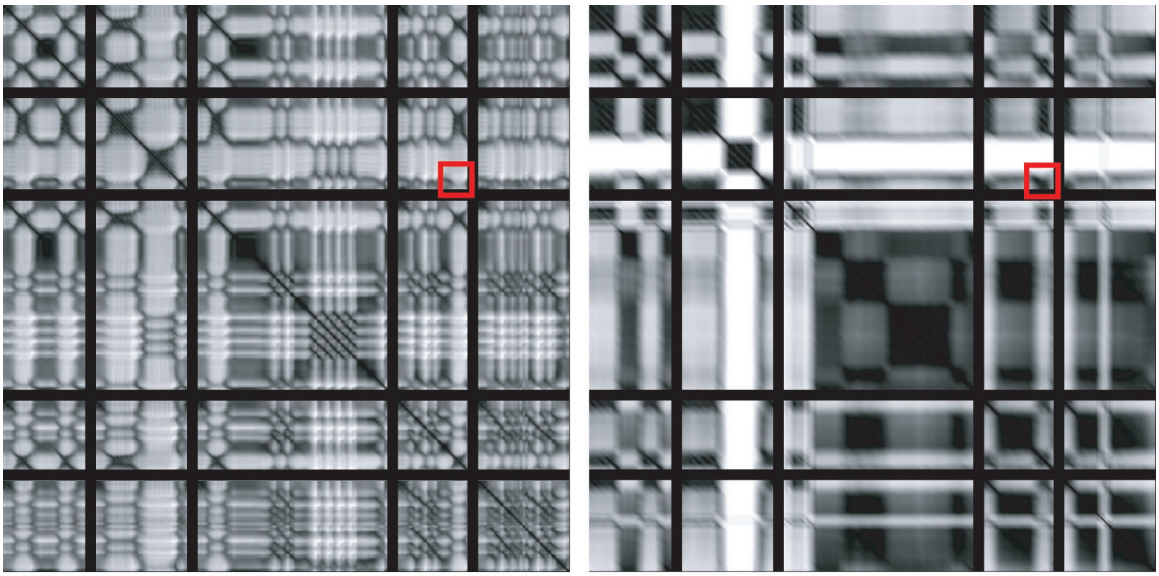


Figure 35: (A) Transition cost matrix computed using figure silhouette alone. **(B)** Cost matrix computed from a traditional motion graph distance metric. Black bars indicate clip boundaries in time. Note how many local minima are evident in (A) in comparison to (B). Also, note that there are low cost regions in matrix (B) that correspond to relatively high costs in matrix (A) (e.g. red highlighted region).

A plausible human video texture should only contain transition images generated

from similar pose sequences. However, robust tracking of limbs and loose clothing while accounting for self-occlusions remains an open problem. As an improvement on the SSD metric used in video textures, we computed a similarity matrix based on the figure silhouette (Fig. 35A). Each matrix entry is the sum of matte pixels following rigid registration (using silhouette centroid and height) and XOR intersection, aggregated over 1.5 seconds of video. Figure 35 illustrates the differences between a silhouette-based and motion capture-based cost matrix (which represents ground truth). Our silhouette-based similarity matrix is clearly inadequate for identifying plausible transitions for human video textures. There are many more local minima in Figure 35A than 35B and a local minima in one matrix does not always correspond to a local minima in the other.

To further complicate issues, the highly structured nature of human movement and our sensitivity in observing them makes small misalignments during transitions immediately visible (e.g. the ghosting artifacts at silhouette boundaries if the limbs are misregistered) as illustrated in Figure 36.

In this chapter, we address the challenge of synthesizing photorealistic human motion by leveraging the complementary characteristics of motion capture data and video. We use motion capture data that is synchronized to our video source to identify candidate transition points in video clips. By leveraging accurate positional information from markers, these metrics succeed where standard video-based ones fail. Once the transitions have been identified, a video-based motion graph (*video graph*) is then constructed by registering, segmenting and blending source video clips to compute transition clips. By utilizing the image-plane projections of motion capture markers as ground control points, we can accurately separate occluding body parts into separate layers and composite layers across transitions in a seamless manner. Finally, animation frames are sequentially generated from a traversal of the video graph by concatenating source and pre-computed transition clips.

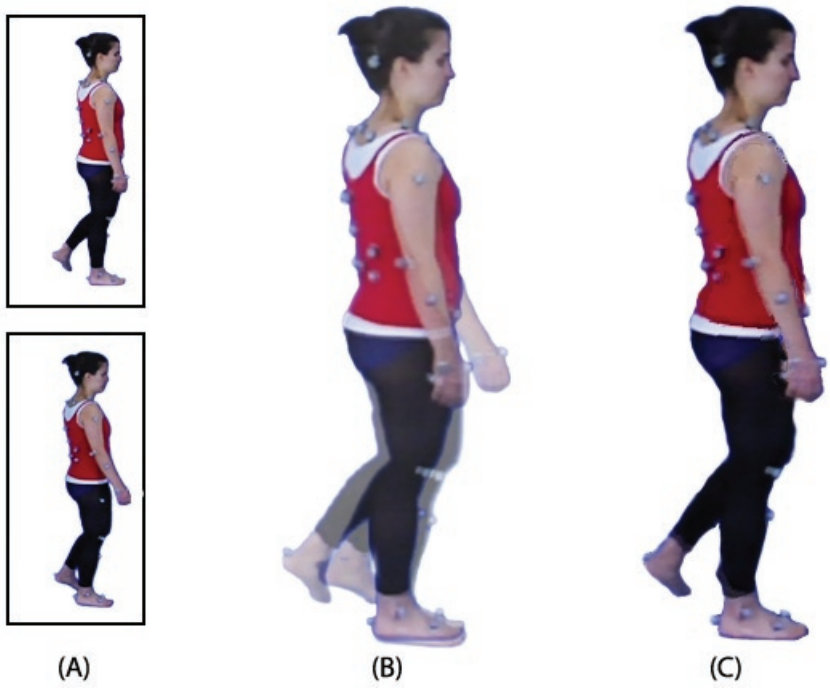


Figure 36: **(A)** One of the challenging goals of transition synthesis is to interpolate two frames from incoming and outgoing video clips. **(B)** Transitions face ghosting artifacts on both the exterior and interior silhouette regions when cross-fading following rigid registration of the figures. **(C)** Our method addresses ghosting outside the silhouette using iterative silhouette deformation and interior ghosting using a novel approach to layered segmentation.

The primary contributions of this chapter are:

- A method for simultaneously capturing marker and video data of human movement and constructing a video graph, which enables the creation of *human video textures*.

- A *technique for synthesizing seamless transitions for human video textures*. This technique uses marker reprojections to control a moving least-squares image warp, and

- A *novel graph cut algorithm for the segmentation of human motion into layers* which exploits marker flow and a superpixel representation of the image data.

To demonstrate the viability of our approach at synthesizing choreographed human movement videos, we present several sets of results (Section 5.6) including: (1) gait motions, to test our system in its detection of transitions and rendering motion on the most basic of human movements, and one for which we are most sensitive to irregularities and (2) an extreme martial arts expert with significant self-occlusion, motion in the depth direction and secondary motions of his clothes.

5.2 Related Work

There are three categories of previous work that are related to our goal of video-based synthesis of human movement: (a) novel view synthesis for captured motions, (b) video texture-based human motion synthesis and (c) video synthesis from mocap data.

5.2.1 Novel view synthesis

A number of works capture high-quality representations of human movement, which enable the replay of captured content from novel 3-D viewpoints. Zitnick et al. [107] use a segmentation-based stereo algorithm to generate a two-layer representation of video frames and compute mattes near depth discontinuities. They demonstrate the ability to synthesize high quality in-between views. Another common approach to the capture and reconstruction of human motion in 3-D is based on the multi-viewpoint construction of the visual hull [25, 85]. While all of these works support the free viewpoint replay of captured human motion in 3-D, they do not address the problem of manipulating captured content to resynthesize completely new sequences.

5.2.2 Video textures

Our approach is an extension of video textures [88]. In particular, we relate to the method of video sprites [87], an extension of video textures which supports flexible

control of transitions by the user, resulting in controllable animations. The frame-level similarity measures used in standard video textures limits them to relatively simple nonrigid or stochastic motions such as waving flags or walking hamsters. As we have illustrated in Figure 2, these methods cannot be applied directly to human video content.

In [27], Zordan et. al. describe an extension of video textures for a carefully-chosen subset of human movements. Their approach employs the ratio of width to height of human silhouette bounding boxes as a feature for identifying transitions. This approach can easily identify motions that generate similar ratios, such as a kick from the right leg versus left leg. For a sufficiently large database of movements, however, this metric will not be sufficiently selective and will require extensive manual intervention. Furthermore, their synthesis technique is based on morphed transitions using the bounding box information. In contrast, our method exploits motion capture data and performs layered motion segmentation to more robustly identify transitions and overcome the ghosting artifacts that result from morphing.

5.2.3 Motion graphs

Our use of motion capture data draws from a substantial body of work on data-driven synthesis of 3-D animations of human characters [57]. In this chapter, we propose a representation of captured video and mocap content which we call the *video graph*. This data representation allows us to jointly leverage mocap and video data to synthesize novel human motions with articulations, and nonrigid variations due to the secondary motion of clothes and hair.

5.2.4 Image and Video-based Re-animation of Humans

There are two additional works on image-based animation of human movement which are related to our approach. A technique described in [29] addresses the problem of capturing and animating the fine-scale clothing deformations of a moving arm. In

contrast, we address the re-synthesis of the entire figure. The method of [45] animates a single image of an articulated creature from motion capture data using manually specified correspondences between image features and 3-D motion features. Like our approach, their method handles limb occlusions using a layered representation of the image and hole-filling using in-painting techniques. However, the crucial step of fitting an initial layered mesh to the image is done with manual intervention. Furthermore, their method does not handle video content.

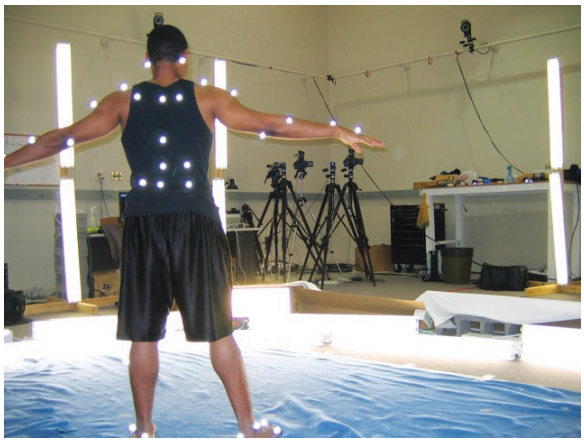
The related work which is perhaps the most similar to ours is the method of Starck et al. [92]. This approach combines reconstruction using a visual hull method with the re-synthesis of human motions. They employ a spherical matching algorithm based on the 3-D mesh of the reconstructed figure to identify good transitions. This approach enables them to construct a type of motion graph using video information alone. However, their technique is limited to genus zero surfaces and relies on accurate 3-D reconstruction. As a consequence, it tends to smooth geometric features (such nose, ears, hair, and clothing) due to reconstruction error and limited geometric resolution. In contrast, our goal is to manipulate source pixels directly with minimal loss of fidelity. This is necessary to preserve crucial details such as loose hair and the folds in clothing. By supplementing video with motion capture data, we obtain the ability to easily identify transitions and synthesize non-genus zero poses.

5.3 Data capture and calibration

The hardware basis of our setup is a commodity motion capture system and video camera. Our motion capture setup is a 12 camera Vicon system capturing 3-D marker trajectories at 120Hz. Video was recorded using a single Panasonic HVX200 camera capturing 1280 x 720 images at 60Hz. In order to synchronize motion capture video signals in software following capture, actors were asked to clap twice in a characteristic manner before and after each performance. These events were used to solve for a

simple scale and translation transformation. Because the motion capture frames were captured at twice the sampling rate of the video, motion capture data was accurately resampled in time for each frame of video.

(a)



(b)

Figure 37: (a) Capture volume with blue screen and mocap cameras. (b) Fluorescent lighting and co-located mocap and video cameras. Note the lack of a standard mocap suit on the subject, and the use of mocap markers attached to clothing as well as skin.

For calibration purposes, a custom calibration target was designed to strobe infrared and visible green light in lockstep with the motion capture cameras. The calibration target is a lit ping pong ball mounted at the tip of a wand. In the video camera images, a green LED inside the ball causes it to appear as a bright green spot

against a black background. It can be automatically segmented from the video frames using a simple thresholding operation in HSV color space. The ball also contains an IR LED and appears as a single large marker which can be localized in 3-D using the standard Vicon video processing pipeline. The corresponding 2-D projections of each 3-D ball position are obtained through color space analysis of the video frames. Using this set of normalized 2-D to 3-D correspondences, the camera projection matrix is resectioned using the Gold Standard Algorithm 6.1 described in [41].

Another fundamental issue is to determine, in each frame captured by the video camera, which markers are actually visible. The standard Vicon software can compute the projections of each marker along the camera viewing axis, but since it sees the markers from multiple views it cannot guess the visibility of a particular marker with respect to an arbitrarily-chosen viewing direction. We implement this visibility test in hardware by positioning an extra Vicon camera next to our video camera, in attempt to colocate their viewing axes.

5.4 Identifying Video Graph Transitions

The original motion graphs paper [57] computed transition candidates as a function of clouds of points over a window of frames in time. Likewise, our method computes motion similarity from the markers directly, instead of from a fitted skeleton, by computing the L2 distance of their positions and trajectories from frame to frame over a fixed window of 0.25 seconds worth of 3-D marker positions. We use a fixed threshold on this distance metric and hand-select this metric to balance the quality of the transitions with the number of candidates.

Since we are using a single video camera in our capture setup, there is only one viewing direction into the scene for which we have video data. As a consequence, the construction of the video graph must be constrained to preserve the continuity of motion with respect to this viewing direction. This has two consequences for the

detection of transitions in building the video graph. First, in computing the cost of a potential transition, we align the two candidate clips by pure translation of the marker points, rather than the rigid body rotation and translation which is used in a standard motion graph. This is because we cannot rotate the captured figure in 3-D without extrapolating the effect of that rotation to our video, a process which is unlikely to produce the realism that we desire.

The second difference is that we prune the set of initial 3-D-based transitions by examining only the 2-D projections of the marker data into the video camera plane. Following camera calibration (see Appendix for details), we match clips when they are compatible with respect to the projected positions and velocities (marker flow) of their marker sets in 2-D, not in 3-D.

More specifically, we prune transition candidates if any corresponding marker flows form an angle greater than a threshold (which we set to 90 degrees).

In our Kung Fu dataset, we picked an initial motion graph threshold that generated approximately 100 transition candidates. The following transition pruning step left about 20 transitions in the Kung Fu video graph and we use these in the paper and video results.

5.5 Generating transitions for video graphs

A human video texture is generated by traversing the video graph, replaying the stored clips and previously synthesized transition clips in an analogous manner to a motion graph[1]. Synthesis of natural-looking transitions is the key technical challenge. Any misregistration of the limbs or torso in transitioning between clips will result in highly visible artifacts such as ghosting and popping. In order to generate believable transitions, we must compute the correspondence between the figure in a pair of clips, warp the video frames so as to bring the images of the figures into alignment,

[1]This approach is well-suited for real-time rendering because all frames are computed and stored before run-time.

and blend the two clips together in the transition region. We will show that these video-manipulation tasks can be simplified considerably through the use of marker data.

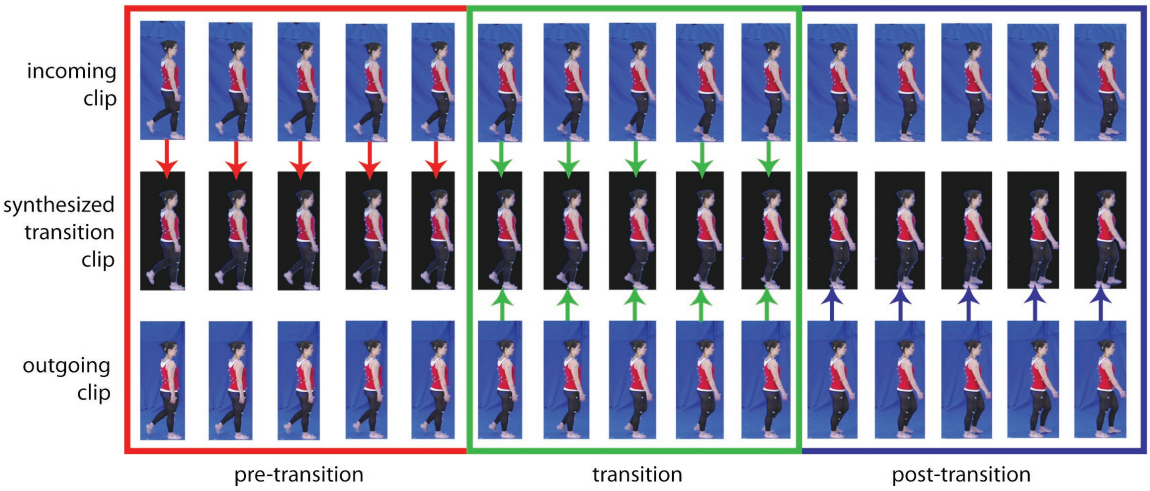


Figure 38: Transition Synthesis: Transition synthesis involves interpolating an incoming clip (top row) with a corresponding outgoing clip captured at a different point in time (bottom row). By linearly interpolating 2-D marker projections from an incoming (or outgoing) frame, a moving least squares warp may be computed to bring marker projections into alignment in the synthesized result (middle row). At the halfway point (8th column), the interpolant value (ranging from $\frac{1}{16}$ to $\frac{15}{16}$) is 0.5 and the warped frames are 50-50 blended. Note that the arm exhibits self-occluding motion – we segment limbs into layers for separate warping and blending from the background layer. Frames in the pre- and post-transition regions (red and blue boxes) are warped and added to the output without blending. This pre- and post-warping reduces warping distortion in the transition region (green box). Arrows denote frames that contribute pixels to the synthesized transition clip.

Transitions are generated over a 15 frame window, as illustrated in Figure 38. The set of synthesized frames are organized into three groups of five frames each, called pre-transition, transition, and post-transition. In the transition region, pixels from both clips are combined to produce the final output. We found that the addition of pre- and post-transition warping resulted in smoother transitions. In these regions there is no blending, and the final output pixels are taken from a single clip (the incoming clip for the pre-transition region and the outgoing clip for post-transition). However, by warping the pre- and post-transition frames towards their corresponding

frames we reduce the amount of warp that is needed in the transition region, which in turn reduces visible artifacts. The total warp that aligns a pair of frames can be broken down into two component warps, one for each frame. These component warps are depicted by arrows in Figure 38.

The key step in the synthesis of a transition is the computation of a pair of warps from a pair of frames. This process can be divided into two conceptual steps: *registration* and *layer segmentation*. The registration step identifies correspondences between the figure regions in both frames and computes the pair of warps, one from each direction. For frames in the transition region, where the magnitude of the warp is the largest and pixels from both clips are blended together, it is important to take into account the fact that different parts of the body (e.g. the arms and torso) may be undergoing self-occluding motions. For example, in the case where the right arm moves across the torso, a single smooth warp for the entire frame will not suffice (the warp would need to "tear" at the occlusion boundary). Column 8 in Figure 38 illustrates such a case: the right hand is outside the leg in the incoming clip and partly inside the leg in the outgoing clip. We address this challenge by automatically segmenting the body parts into layers. In the given example, we segment the right arm pixels from the rest of the body. The arm then comprises a foreground layer while the remaining figure pixels constitute a background layer. We can then compute separate smooth warps for the corresponding segmented layers in a pair of frames and composite the warped layers together in generating the final output.

The complete image synthesis process for a pair of transition frames consists of the steps illustrated in Figure 39. In the first step, rigid translation and scaling of the two video clips produces a coarse alignment between pairs of corresponding frames. Then for each frame, we compute a nonrigid warp using moving least squares (MLS) [86]. The MLS warp uses a set of control points which consist of reprojected markers and a set of point samples from the silhouette boundary. Thus the registration step aligns

the silhouette boundaries between the frames. The third step performs a layer-based decomposition of the input video so that moving limbs can be segmented out and warped separately from the background layer. In the final step, the output clips are blended and composited, and any remaining holes are filled by inpainting, to produce the output clip. We now describe each of these steps in more detail.

5.5.1 Scaled Rigid Registration

First, we roughly register the character regions by adjusting translation and global scale for each pair of corresponding frames. Because the characters' poses can be assumed similar at corresponding frames, we can adopt a simple approach for registering character regions: we first find the "rigid" marker positions, then estimate the translation and scale based on bounding rectangles. A rigid marker is one whose apparent motion is the smallest; it is used to compute the shift. To find the scale (about this rigid marker), we globally scale the bounding boxes to match their heights.

5.5.2 Iterative silhouette deformation

In order to produce an accurate alignment of figure regions from two frames, we proceed in two stages. In the first stage, we begin by identifying the point correspondences corresponding to the projected positions of background layer markers which are visible in both frames.

We do an initial MLS warp with these marker projections as control points. Because we know which body part each marker is attached to, we can trivially assign each visible marker to its corresponding layer.

The initial warp brings the background layer body shapes into approximate alignment, but it is insufficient for full body warping. This is because such control points do not provide explicit information about the boundary. If the boundary is not taken into account, ghosting invariably occurs in the interpolated frames. Therefore, in the second stage we perform additional warping of the two body shapes in order to bring

their silhouette boundary curves into alignment.

Our alignment method is essentially an application of the Iterated Closest Point algorithm [15]. It has the advantages of being simple to implement and very effective for curves which are already in close proximity to each other. This will always be the case in our application due to the reliability of the initial warp using the projected markers. We apply additional silhouette alignment constraints to the MLS warp by sampling control points from the silhouette curves and incrementally updating the MLS solution. We can measure the degree of overlap between body shapes by computing the Sum of Absolute Differences (SAD) between the warped mattes in the two frames. This measure counts the number of pixels that lie outside the overlap region. We identify point correspondences between contours that minimize the SAD error measure and add them to the control point set for the MLS estimate. After reaching convergence, the intersection of the registered mattes is computed and used for final composition to eliminate remaining exterior ghosting artifacts (Fig. 39 shows the registered result before matte intersection to illustrate how ghosting outside the silhouette is reduced before it is eliminated with intersection).

5.5.3 Layered motion segmentation

A key step in the accurate treatment of self-occluding motion (e.g., when an arm moves in front of the torso) is to decompose the figure region into layers that can be warped separately. In general, the problem of automatically decomposing an arbitrary video sequence into an appropriate set of layers is quite challenging and has received considerable attention. In our application, we can leverage the context and motion estimates obtained from our reprojected marker set to solve for the layer segmentations using a novel application of MRF labeling via graph cuts.

The first step in our process is to segment the figure pixels in each frame, resulting in a set of pixel regions known as superpixels. We use the method of [33] to generate

the superpixels. We then generate a Markov Random Field (MRF) [67] model for the segmentation problem, where the observation nodes are superpixels and the hidden labels are either foreground or background. After performing MRF inference via graph cuts (we use the method of [20]), we assign the label for each superpixel to all of the image pixels that it contains. The superpixel approach reduces the number of nodes that are required in the MRF, thereby reducing the memory and computation requirements.

Our assignment of edge costs in the graph is illustrated in Figure 40 and specified in Equation 4 below. We leverage the initial warps obtained from the limb-labeled marker projections to separate the foreground and background layers effectively. In this example there is one foreground layer corresponding to the arm and one background layer. Assume we are warping from outgoing frame B to incoming frame A. We first compute two separate warps of image B based upon the foreground and background visible marker sets. The results of applying these warps to the image B are illustrated in the middle column of Figure 40. The cost of associating a superpixel in image A with the background (torso) label is computed by comparing the superpixel color with the corresponding region in the background-warped output (Fig. 40(3)). The cost for a foreground layer assignment is similarly computed using the foreground warped output (Fig. 40(2)).

More specifically, we minimize the following energy function over all candidate binary labelings L of image A:

$$E(L) = \sum_{p \in A} \left(\frac{1}{\mid p \mid} \sum_{i \in p} \| A_i - B_i^{'}(L_p) \|^2 \right) + \beta \sum_{pq \in N} T\left(L_p \neq L_q\right) \cdot \exp\left(-\|\overline{A_p} - \overline{A_q}\|^2 / 2\sigma^2\right) \tag{4}$$

where p is a superpixel, A_i is the RGB color of pixel i in image A and $B_i^{'}(L_p)$

is the color of the corresponding pixel in image B under the MLS warp specified by label L_p. In the third summation term, which penalizes discontinuities between neighboring superpixels $pq \in N$ with similar color, function $T(L_p \neq L_q)$ is 1 if the condition inside parenthesis is true and 0 otherwise. Also in the third summation, β is a design parameter (which we set to 30 in our experiments), $\overline{A_p}$ and $\overline{A_q}$ are the mean colors of superpixels p and q respectively and σ is a noise parameter.

The intuition behind this formulation is that superpixels that belong to a particular layer in one frame of a pair will be well-matched to the pixels in the corresponding frame, once the layer-specific warping has been performed. Note that if the relative motion between foreground and background is very small, or if the foreground and background colors are very similar, it may not be possible to segment the layers accurately. Fortunately, in that case an accurate segmentation is not needed since ghosting artifacts will only be visible if there is a color mismatch. Otherwise, when the wrong warp is applied it is quite unlikely that the superpixel will find significant support in the image pixel values. In general we deal with multiple possible foreground layers by doing a series of binary segmentations, one for each possible foreground. In each case we used the limb-labeled marker sets to compute the appropriate warps. Following graph cut segmentation for each layer, we enforce segmentation consistency between image A (B) and B (A) by eliminating foreground-labeled superpixels in image A (B) that do not overlap by a threshold amount (0.3) in area with corresponding foreground-labeled superpixels in image B' (A').

5.5.4 Rendering transition frames

Given a set of segmented and warped layers for each frame in the transition region from each video clip, we perform compositing and hole-filling to render the final video sequence. First we add the pre- and post-transition frames to the output sequence by applying a global warp to each frame. Then we composite the segmented layers

from each pair of frames in the transition region. We use the painters algorithm and composite the layers from back to front (i.e. background layer followed by foreground). There will be a separate foreground layer for each MRF segmentation. We use the average distance from the camera of the mocap markers in each layer to determine their order.

A significant problem in compositing the foreground layers is the presence of holes due to missing background pixels. For example, in order to align the right arm across a pair of images, we may need to shift the arm up slightly in one frame and down in the other. These shifts will create holes at the trailing edge of the warp because we are uncovering background layer pixels which are not present in the source imagery. We address this problem in two ways. First, we differentiate between overlapping and nonoverlapping pixels in compositing the layers. Overlapping pixels belong to the intersection of the corresponding foreground layers from both frames (incoming and outgoing), while nonoverlapping pixels belong to one layer and not the other. We crossfade overlapping pixels in order to generate a smooth transition in appearance. We then composite the nonoverlapping pixels directly into the output buffer. This allows us to use all of our foreground layer pixels and minimizes the number of background holes that must be filled.[2] Finally, any remaining holes in the background layer are filled automatically using in-painting [32].

5.6 Results

In this section, we present results from applying our method to three different datasets resulting from three capture sessions with two different subjects. For each dataset, we constructed the video graph and generated synthetic video sequences. The full sequences are available in the video accompanying this paper. In this section we highlight specific transitions from these sequences in order to illustrate some of the

[2]This strategy does have the property of making the foreground layers "fatter," but it does not seem to be a significant source of artifacts in practice.

results of our method.

The first result consists of backflips performed by an acrobatic martial artist. The second result features the same martial artist performing a range of Kung Fu fighting moves. The third result exhibits female gait motion.

5.6.1 Martial arts demonstration

Figure 41 shows three example transition composites from a martial arts expert captured wearing two separate costumes: (a) flowing black shorts and (b) baggy red pants. Our system identified a transition during punches toward the camera from a set of captured backflips. This is an example of a non-trivial transition that would be difficult to identify by hand. Despite significant self-occluding motion from his left arm, the transition result displayed Figure 41(C) shows his hand in focus. One side effect of our method leaves neighboring pixels to a moving layer blurry, which is caused by a combination of hole removal and non-overlapping regions of the layer.

5.6.2 Gait motion

We also captured a woman performing simple walks which serve as tough examples since people are used to observing gait. Normal gait also provides clear sources of self-occlusion as people swing their arms. As evidenced in Fig. 41(B) and 41(E), our layered segmentation method significantly reduces texture mismatch caused by the arms moving beside the body.

We have chosen to leave the visible markers in all but one of the results we present in this paper. This facilitates the comparison between frames and ensures that any visible artifacts are the result of our core algorithm. In practice, marker removal would be performed on each frame to generate the final output and we show an example of this manual correction in Fight Sequence 2 in the included video. This is a standard operation in many production scenarios [17] and commercial software is available to facilitate it.

5.7 Limitations

The key challenge in synthesizing photo-realistic human motion from video is the need to establish correspondences between regions of pixels across the video sequence. These correspondences are needed to (a) estimate the extent of motion between frames (to identify potential transitions) and (b) to segment, warp, and align images between frames to create seamless transitions. Standard computer vision methods for video analysis are unable to reliably identify the correct correspondences with sufficient accuracy to support our application.

Our solution to the correspondence problem is to leverage motion capture technology to obtain accurate and reliable correspondence information in the form of marker data. While we believe this approach has enabled us to obtain synthesis results which would be difficult to achieve through any other method, it is worthwhile to discuss the practical limitations of our current solution. One set of issues center around the capture environment, which must be tailored to the conflicting needs of video- and motion-capture. Costume design and marker placement must be carefully thought-out. Clothing was tightened and adjusted appropriately to ensure marker visibility and stability during the sessions. The motion of the actors themselves was not choreographed in a precise manner, but they were certainly aware of the need to conform to the motion style of a side-scrolling video game.

Because we currently employ a single video camera, we are sensitive to parallax effects (parallax can be observed, for example, in the relative motion of the shoulders with respect to the center of the chest). Fortunately there has been significant progress in multi-camera technology for 3-D capture of human motion, and we believe that we can leverage these results to extend our system's capabilities to 3-D. This would also enable 3-D visualizations of newly synthesized content and open up additional application domains. Another issue that we plan to address in future work is relighting the synthesized content to reduce changes of intensity across transitions, which is

evident in the results, and animation in new environments.

5.8 Discussion

We found that transition video clips may be easily noticed in the presence of ghosting artifacts and discontinuities in motion. Our system's exploitation of marker data in addition to captured pixels was vital for the elimination of these artifacts. While our method is capable of generating transition clips which are difficult to detect during video playback, careful examination of still transition frames reveals subtle artifacts such as blurriness and non-smooth layer boundaries that expose their synthetic nature.

Also, as in the case of motion graphs, we identified some transitions that were surprising in the sense that they could not have been easily predicted by a capture session director. For example, the two transitions composed using transition frames shown in Fig. 7(A) and 7(C) would have been difficult to identify by hand. Therefore, the transition discovery capabilities of motion graphs carry over to the video domain in our computation of video graphs.

A number of interesting extensions to our work are possible. First, our layered motion segmentation could be improved by adding a second video camera and incorporating additional stereo cost terms in the MRF. This would introduce depth discontinuity information in finding limb boundaries in addition to the information provided by oversegmentation (in a similar fashion to [107]). Second, our method faces the challenges of trading transition quality for motion responsiveness (graph connectivity) which is common to all motion graph-like interactive animation systems. By expanding motion and video data via interpolation of subsequences between similar foot placement events [105], more transitions may be introduced to improve character responsiveness to interaction. Finally, we are excited about the possibilities

for re-animating videos of humans using separate motion capture of alternate characters. The technique of [45] for animating a still picture using motion capture data could be extended to video using the methods presented in this paper.

5.9 Summary of the chapter

In this chapter, we have presented a method for creating controllable photorealistic animations of human movement. By capturing video and motion capture data in tandem, we have demonstrated that video clips of similar pose sequences from different points in time may be identified from 3D and 2D projected marker trajectories. We have shown how to render synthetic video clips for transitioning video across gaps in time via novel registration and segmentation algorithms.

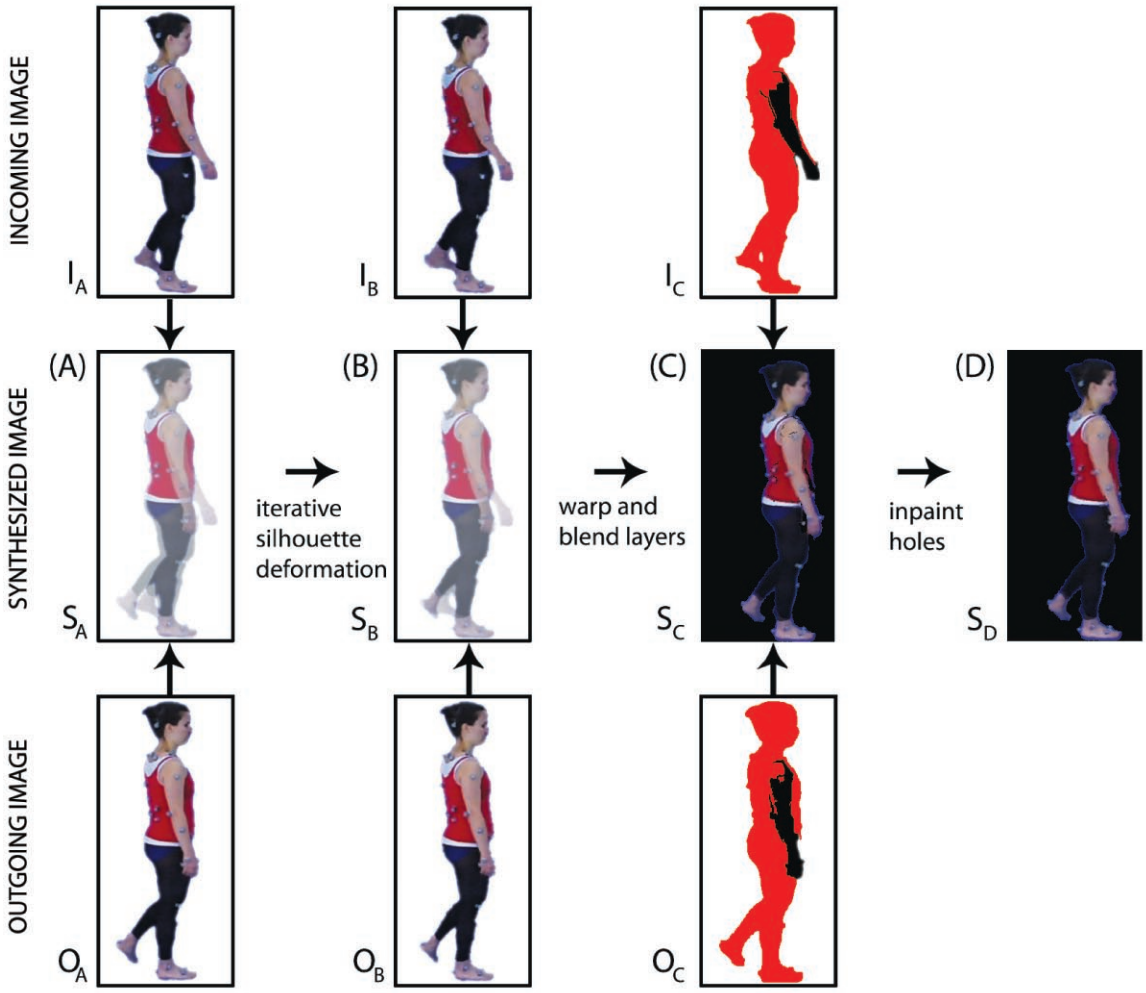


Figure 39: Transition Image Synthesis Pipeline: The following steps generate output S_D from incoming image I_A and outgoing image O_A: (A) I_A and O_A are rigidly registered to align root marker projections and silhouette height, producing S_A. (B) Iterative silhouette deformation is applied to I_A and O_A, producing I_B and O_B - note the reduced ghosting behind the legs in S_B from S_A. (C) I_A and O_A are segmented into motion (limb) layers which are warped, blended and composited onto the background layer in back to front order, producing S_C. (D) Finally, image in-painting is applied to S_C to fill holes between composited layers, resulting in S_D.

:

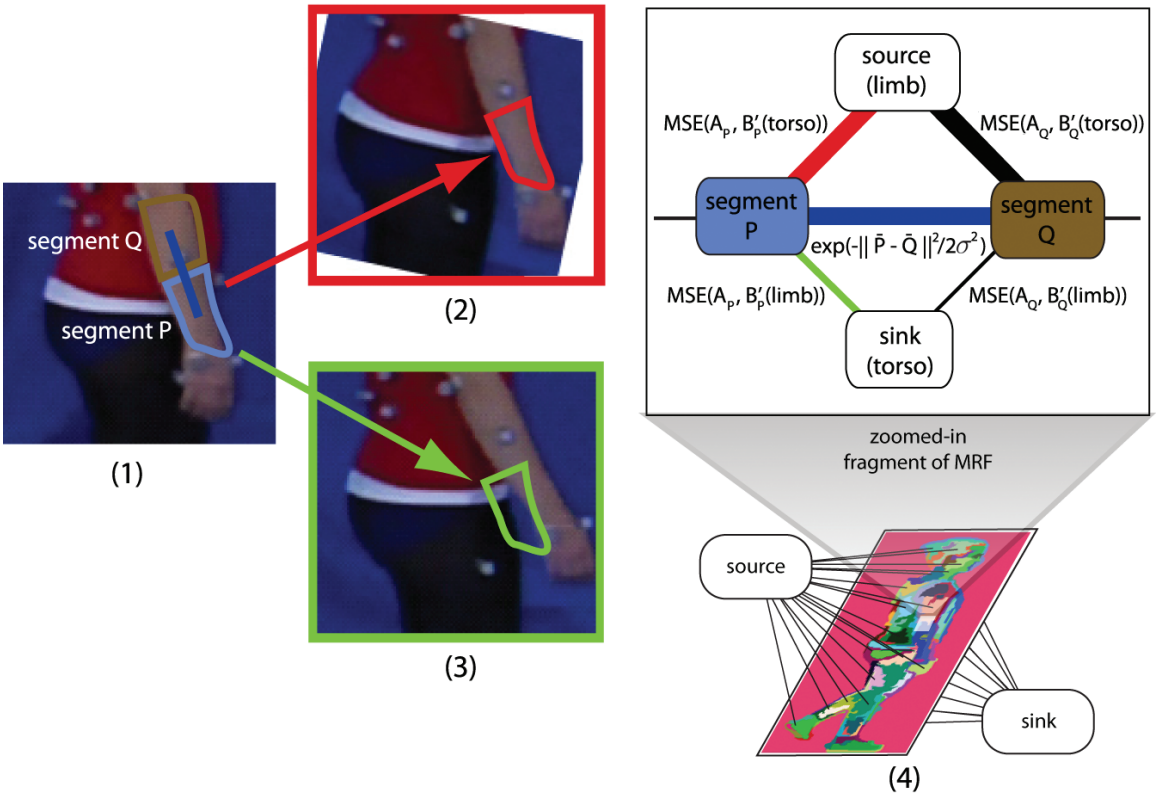


Figure 40: Graph cut based motion layer segmentation: In this example, image A represents the image to segment and image B represents the corresponding image in the transition pair. **(1)** Image A with cyan superpixel P to be labeled as foreground (limb) or background (torso). **(2)** Image B is warped towards image A using foreground (limb) markers, producing $B^{'}(limb)$, **(3)** Image B is warped toward image A using background (torso) markers, producing $B^{'}(torso)$. **(4)** Fragment of MRF model showing the organization of the cost terms, where thick lines denote high capacity. Following graph cut, superpixels connected to the source (sink) are labeled as foreground (background).

Figure 41: Transition Composites: Images in the top row show transition frames cross-faded after rigid registration. Note the ghosting artifacts inside and outside the figures' overlapping silhouettes. The bottom row shows the result of applying our transition synthesis method.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this dissertation, we have explored several techniques for synthesizing crowds from imagery. Video-based crowds are capable of depicting natural individual and group behaviors in a photorealistic manner because they are composed from examples captured from the output scene. In contrast, current techniques for synthesizing crowds for special effects, games and architectural visualization are accomplished via a complex and labor-intensive model-based graphics pipeline.

We have presented a representation of segmented video objects called *crowd tubes* which enable satisfaction of constraints on the motion and appearance of video billboards imposed by the 3D scene. In our experiments, crowd tubes were constructed and used to layout crowds which were recorded from the real world. Video-based crowd effects can therefore benefit from the recent proliferation of large format cameras such as those from the Red company [5]. Moreover, crowd tubes may also be applied to modify crowd video generated with traditional computer graphics. This could potentially be very useful to visual effects producers which currently rely on a slow and tedious model-based pipeline for synthetic crowd production.

Several video-based crowds were composed using a prototype system for authoring constraints, instantiating crowd tubes and animating and rendering the resulting 3D composition of video billboards. In addition to providing a range of crowd outputs which demonstrate the capabilities of the prototype system, a qualitative user evaluation revealed several issues and directions for improvement with the crowd authoring interface. This dissertation also presented a novel technique for segmenting video objects that are temporally coherent and a system and pipeline for constructing human

video textures, an extension of video textures to human motion.

There are several exciting avenues for future work in video-based crowd synthesis:

- Large databases of high quality blue screen-matted video clips are becoming commercially available, presenting opportunities for large scale video-based crowd synthesis. This presents several new challenges including how to automatically transfer matted clips across scenes while accounting for variations in lighting and appearance. It also presents great computational demands on the present constraint satisfaction approach, which is NP-hard.

- A design gallery-based approach to authoring crowds could enable a higher level approach to controlling captured behaviors at the individual and macro scales. This would demand more rapid techniques for animating and rendering crowds than those presented in this dissertation. As discovered during the qualitative user evaluation (Sec. 3.5), user-controlled behavior grouping and categorization would potentially ease frustrations with the crowd authoring process. A design gallery-based approach could easily take advantage of user groupings by permuting layouts within each behavioral category (e.g. moving vs. static pedestrian galleries).

- Real-time video-based crowd synthesis would enable visually exciting crowd effects for the emerging category of *cinematic gaming*.

- Crowd synthesis packages, such as Massive [3], could incorporate the crowd tube representation and optimization technique for rapid re-use of previously animated and rendered behaviors.

- Crowd tubes could be merged into supernodes representing crowd layout templates, such as lane-forming and bottlenecking animations.

- Free viewpoint camera control based on multi-view crowd capture would pose novel challenges to video-based crowd synthesis, especially in the face of a large number of occlusions. Image completion techniques could potentially be incorporated into our constraint satisfaction framework.

- Identification and representation of additional constraints on a crowd's composition, informed by ongoing perceptual studies, could improve realism.

REFERENCES

[1] "UCF Sports Action Dataset." http://server.cs.ucf.edu/ vision/data.html.

[2] "The Matrix (film), Warner Bros.," 1999.

[3] "Massive, crowd animation software for visual effects, http://www.massivesoftware.com," 2003.

[4] "Grand Theft Auto, Rock Star Games, http://www.rockstargames.com/iv/," 2009.

[5] "Red One, http://www.red.com," 2009.

[6] "SimCity, http://simcity.ea.com," 2009.

[7] Agarwala, A., Zheng, K. C., Pal, C., Agrawala, M., Cohen, M., Curless, B., Salesin, D., and Szeliski, R., "Panoramic video textures," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, (New York, NY, USA), pp. 821–827, ACM Press, 2005.

[8] Agarwala, A., Zheng, K. C., Pal, C., Agrawala, M., Cohen, M., Curless, B., Salesin, D., and Szeliski, R., "Panoramic video textures," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, (New York, NY, USA), pp. 821–827, ACM, 2005.

[9] Andrade, D., Resende, M., and Werneck, R., "Fast local search for the maximum independent set problem," in *Proceedings of 7th International Workshop on Experimental Algorithms (WEA 2008)*, 2008.

[10] Avidan, S., "Support vector tracking," *IEEE Trans. PAMI*, vol. 26, no. 8, pp. 1064–1072, 2004.

[11] Bai, X., Wang, J., Simmons, D., and Sapiro, G., "Video snapcut: Robust video object cutout using localized classifiers," in *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, (New York, NY, USA), pp. 1–11, ACM, 2009.

[12] Ballan, L., Brostow, G. J., Puwein, J., and Pollefeys, M., "Unstructured video-based rendering: interactive exploration of casually captured videos," in *SIGGRAPH '10: ACM SIGGRAPH 2010 papers*, (New York, NY, USA), pp. 1–11, ACM, 2010.

[13] Bayazit, O. B., ming Lien, J., and Amato, N. M., "Better group behaviors in complex environments using global roadmaps," in *In Artif. Life*, pp. 362–370, 2002.

[14] Beier, T. and Neely, S., "Feature-based image metamorphosis," *In Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, vol. 26, no. 2, pp. 35–42, 1992.

[15] Besl, P. J. and McKay, N. D., "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[16] Bibby, C. and Reid, I., "Robust real-time visual tracking using pixel-wise posteriors," in *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, (Berlin, Heidelberg), pp. 831–844, Springer-Verlag, 2008.

[17] Borshukov, G., Hable, J., and Montgomery, J., *Playable Universal Capture in GPU Gems 3.* Addison Wesley, 2007.

[18] Bose, B. and Grimson, E., "Ground plane rectification by tracking moving objects," in *IEEE International Workshop on Visual Surveillance and PETS*, 2004.

[19] Bouguet, J.-Y., "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," 2000.

[20] Boykov, Y. and Kolmogorov, V., "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[21] Boykov, Y. and Funka-Lea, G., "Graph cuts and efficient n-d image segmentation.," *International Journal of Computer Vision (IJCV)*, vol. 70, no. 2, pp. 109–131, 2006.

[22] Boykov, Y., Veksler, O., and Zabih, R., "Fast approximate energy minimization via graph cuts," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 11, pp. 1222–1239, 2001.

[23] Brostow, G. J. and Cipolla, R., "Unsupervised bayesian detection of independent motion in crowds," in *IEEE Computer Vision and Pattern Recognition*, pp. I: 594–601, 2006.

[24] C. Rother, V. Kolmogorv, A. B., "Grabcut: Interactive foreground extraction using iterated graph cuts," *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, vol. 23, no. 3, pp. 309–314, 2004.

[25] Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H.-P., "Free-viewpoint video of human actors," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 569–577, 2003.

[26] Caselles, V., Kimmel, R., and Sapiro, G., "Geodesic active contours," *IJCV*, vol. 22, no. 1, pp. 61–79, 1997.

[27] Celly, B. and Zordan, V., "Animated people textures," in *17th International Conference on Computer Animation and Social Agents*, 2004.

[28] Chockalingam, P., Pradeep, N., and Birchfield, S. T., "Adaptive fragments-based tracking of non-rigid objects using level sets," in *ICCV*, 2009.

[29] Cobzas, D., Yerex, K., and Jgersand, M., "Dynamic textures for image-based rendering of fine-scale 3d structure and animation of non-rigid motion," in *Eurographics*, pp. 1067–7055, 2002.

[30] Cohen, M. F., Shade, J., Hiller, S., and Deussen, O., "Wang tiles for image and texture generation," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 287–294, 2003.

[31] de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H., and Thrun, S., "Performance capture from sparse multi-view video," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 4:1–4:10, 2008.

[32] Efros, A. A. and Leung, T. K., "Texture synthesis by non-parametric sampling," in *ICCV (2)*, pp. 1033–1038, 1999.

[33] Felzenszwalb, P. F. and Huttenlocher, D. P., "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[34] Ferwerda, J. A., Shirley, P., Pattanaik, S. N., and Greenberg, D. P., "A model of visual masking for computer graphics," in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 143–152, ACM Press/Addison-Wesley Publishing Co., 1997.

[35] Flagg, M., Nakazawa, A., Zhang, Q., Kang, S. B., Ryu, Y. K., Essa, I., and Rehg, J. M., "Human video textures," in *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, (New York, NY, USA), pp. 199–206, ACM, 2009.

[36] Frey, B. J. and Dueck, D., "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[37] Funge, J., Tu, X., and Terzopoulos, D., "Cognitive modeling: knowledge, reasoning and planning for intelligent characters," in *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 29–38, ACM Press/Addison-Wesley Publishing Co., 1999.

[38] Ge, W. and Collins, R., "Marked point processes for crowd counting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR09)*, pp. 1–8, 2009.

[39] Goldman, D. B., Gonterman, C., Curless, B., Salesin, D., and Seitz, S. M., "Video object annotation, navigation, and composition," in *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, (New York, NY, USA), pp. 3–12, ACM, 2008.

[40] Haritaoglu, I., Harwood, D., and Davis, L. S., "W4: Real-time surveillance of people and their activities," *IEEE Trans. PAMI*, vol. 22, no. 8, pp. 809–830, 2000.

[41] Hartley, R. I. and Zisserman, A., *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[42] Helbing, D., Molnar, P., Farkas, I. J., and Bolay, K., "Self-organizing pedestrian movement," *Environment and Planning B: Planning and Design*, vol. 28, pp. 361–383, 2001.

[43] Helbing, D., Buzna, L., Johansson, A., and Werner, T., "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," *Transportation Science*, vol. 39, no. 1, pp. 1–24, 2005.

[44] Hiebert, B., Dave, J., Kim, T.-Y., Neulander, I., Rijpkema, H., and Telford, W., "The chronicles of narnia: the lion, the crowds and rhythm and hues," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, (New York, NY, USA), p. 1, ACM, 2006.

[45] Hornung, A., Dekkers, E., and Kobbelt, L., "Character animation from 2d pictures and 3d motion data.," *ACM Transactions on Graphics*, vol. 26, no. 1, 2007.

[46] Isard, M. and Blake, M., *Active Contours*. Springer, 1998.

[47] Isard, M. and MacCormick, J., "BraMBLe: A Bayesian multiple-blob tracker," in *Proc. ICCV*, vol. 2, pp. 34–41, 2001.

[48] Jiangjian Xiao, Shah, M., "Motion layer extraction in the presence of occlusion using graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 10, pp. 1644–1659, 2005.

[49] Ju, E., Choi, M. G., Park, M., Lee, J., Lee, K. H., and Takahashi, S., "Morphable crowds," *ACM Trans. Graph.*, vol. 29, 2010.

[50] Junejo, I. and Foroosh, H., "Trajectory rectification and path modeling for video surveillance," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1–7, 2007.

[51] Kao, E. K., Daggett, M. P., and Hurley, M. B., "An information theoretic approach for tracker performance evaluation," in *ICCV*, 2009.

[52] Kavan, L., Dobbyn, S., Collins, S., Zara, J., and O'Sullivan, C., "Polypostors: 2d polygonal impostors for 3d crowds," in *2008 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 149–155, ACM Press, February 2008.

[53] Klau, G. W., Lesh, N., Marks, J., and Mitzenmacher, M., "Human-guided tabu search," in *In Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 41–47, 2002.

[54] Kohli, P. and Torr, P., "Efficiently solving dynamic markov random fields using graph cuts," *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 922–929, 2005.

[55] Komodakis, N., Paragios, N., and Tziritas, G., "Mrf optimization via dual decomposition: Message-passing revisited," in *ICCV*, pp. 1–8, 2007.

[56] Kopf, J., Cohen-Or, D., Deussen, O., and Lischinski, D., "Recursive wang tiles for real-time blue noise," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, vol. 25, no. 3, pp. 509–518, 2006.

[57] Kovar, L., Gleicher, M., and Pighin, F., "Motion graphs," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 473–482, 2002.

[58] Kwatra, V., Schdl, A., Essa, I., Turk, G., and Bobick, A., "Graphcut textures: Image and video synthesis using graph cuts," *ACM Transactions on Graphics, SIGGRAPH 2003*, vol. 22, pp. 277–286, July 2003.

[59] Lai, K. K., Xue, J., and Xu, B., "Container packing in a multi-customer delivering operation," *Comput. Ind. Eng.*, vol. 35, no. 1-2, pp. 323–326, 1998.

[60] Lalonde, J.-F. and Efros, A. A., "Using color compatibility for assessing image realism," in *IEEE International Conference on Computer Vision*, 2007.

[61] Lalonde, J.-F., Hoiem, D., Efros, A. A., Rother, C., Winn, J., and Criminisi, A., "Photo clip art," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, August 2007.

[62] Lee, K. H., Choi, M. G., Hong, Q., and Lee, J., "Group behavior from video: a data-driven approach to crowd simulation," in *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, (Aire-la-Ville, Switzerland, Switzerland), pp. 109–118, Eurographics Association, 2007.

[63] Lee, K. H., Choi, M. G., and Lee, J., "Motion patches: building blocks for virtual environments annotated with motion data," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 898–906, 2006.

[64] Lempitsky, V. and Boykov, Y., "Global optimization for shape fitting," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[65] Lerner, A., Chrysanthou, Y., and Lischinski, D., "Crowds by example," *Computer Graphics Forum (Proceedings of Eurographics)*, vol. 26, no. 3, 2007.

[66] Leventon, M. E., Grimson, W. E. L., and Faugeras, O. D., "Statistical shape influence in geodesic active contours," in *Proc. CVPR*, pp. 1316–1323, 2000.

[67] Li, S. Z., *Markov random field modeling in computer vision*. London, UK: Springer-Verlag, 1995.

[68] Li, Y., Huang, C., and Nevatia, R., "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR09)*, pp. 1–8, 2009.

[69] Li, Y., Sun, J., and Shum, H.-Y., "Video object cut and paste," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 595–600, 2005.

[70] Loscos, C., Marchal, D., and Meyer, A., "Intuitive crowd behaviour in dense urban environments using local laws," in *TPCG '03: Proceedings of the Theory and Practice of Computer Graphics 2003*, (Washington, DC, USA), p. 122, IEEE Computer Society, 2003.

[71] Luby, M., "A simple parallel algorithm for the maximal independent set problem," in *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 1–10, ACM, 1985.

[72] Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., and Shieber, S., "Design galleries: a general approach to setting parameters for computer graphics and animation," in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 389–400, ACM Press/Addison-Wesley Publishing Co., 1997.

[73] McDonnell, R., Larkin, M., Dobbyn, S., Collins, S., and O'Sullivan, C., "Clone attack! perception of crowd variety," in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, (New York, NY, USA), pp. 1–8, ACM, 2008.

[74] McDonnell, R., Larkin, M., Hernández, B., Rudomin, I., and O'Sullivan, C., "Eye-catching crowds: saliency based selective variation," in *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, (New York, NY, USA), pp. 1–10, ACM, 2009.

[75] Metoyer, R. A. and Hodgins, J. K., "Reactive pedestrian path following from examples," in *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, (Washington, DC, USA), p. 149, IEEE Computer Society, 2003.

[76] Niebles, J. C., Han, B., Ferencz, A., and Fei-Fei, L., "Extracting moving people from internet videos," in *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, (Berlin, Heidelberg), pp. 527–540, Springer-Verlag, 2008.

[77] Ondřej, J., Pettré, J., Olivier, A.-H., and Donikian, S., "A synthetic-vision based steering approach for crowd simulation," *ACM Trans. Graph.*, vol. 29, pp. 123:1–123:9, July 2010.

[78] Paragios, N. and Deriche, R., "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans PAMI*, vol. 22, pp. 266–280, 2000.

[79] Pellegrini, S., Ess, A., Schindler, K., and van Gool, L., "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV*, 2009.

[80] Price, B. L., Morse, B. S., and Cohen, S., "Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *ICCV*, 2009.

[81] Ren, X. and Malik, J., "Tracking as repeated figure/ground segmentation," in *CVPR*, 2007.

[82] Reynolds, C., "Big fast crowds on PS3," in *Sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, (New York, NY, USA), pp. 113–121, ACM, 2006.

[83] Reynolds, C. W., "Flocks, herds and schools: A distributed behavioral model," in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 25–34, ACM, 1987.

[84] Rodriguez, M., Ahmed, J., and Shah, M., "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8, June 2008.

[85] Sand, P., McMillan, L., and Popovic, J., "Continuous capture of skin deformation," in *ACM Transactions on Graphics*, vol. 22, pp. 578–586, 2003.

[86] Schaefer, S., McPhail, T., and Warren, J., "Image deformation using moving least squares," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 533–540, 2006.

[87] Schödl, A. and Essa, I. A., "Controlled animation of video sprites," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pp. 121–127, 2002.

[88] Schödl, A., Szeliski, R., Salesin, D. H., and Essa, I., "Video textures," in *Proceedings of ACM SIGGRAPH 2000*, pp. 489–498, ACM Press / ACM SIGGRAPH, 2000.

[89] Schoenemann, T. and Cremers, D., "Globally optimal shape-based tracking in real-time," in *Proc. CVPR*, 2008.

[90] Shao, W. and Terzopoulos, D., "Autonomous pedestrians," *Graph. Models*, vol. 69, no. 5-6, pp. 246–274, 2007.

[91] Shi, J. and Tomasi, C., "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[92] Starck, J. and Hilton, A., "Surface capture for performance-based animation," *IEEE Comput. Graph. Appl.*, vol. 27, no. 3, pp. 21–31, 2007.

[93] Sung, M., *Scalable, controllable, efficient and convincing crowd simulation*. PhD thesis, Madison, WI, USA, 2005. Supervisor-Gleicher, Michael L.

[94] Sung, M., Kovar, L., and Gleicher, M., "Fast and accurate goal-directed motion synthesis for crowds," in *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, (New York, NY, USA), pp. 291–300, ACM, 2005.

[95] Tang, A., Greenberg, S., and Fels, S., "Exploring video streams using slit-tear visualizations," in *Proceedings of Advanced Visual Interfaces (AVI'08)*, (Napoli, Italy), pp. 191–198, May 28-30 2008. See related video in Report 2008-897-10. Earlier version published as Report 2007-886-38 in a paper and video, December.

[96] Tecchia, F., Loscos, C., and Chrysanthou, Y., "Image-based crowd rendering," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 36–43, 2002.

[97] Treuille, A., Cooper, S., and Popović, Z., "Continuum crowds," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, (New York, NY, USA), pp. 1160–1168, ACM Press, 2006.

[98] Tsai, D., Flagg, M., and Rehg, J. M., "Motion coherent tracking with multi-label mrf optimization," *British Machine Vision Conference (Best Student Paper)*, 2010.

[99] Vaswani, N., Tannenbaum, A., and Yezzi, A., "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Trans. PAMI*, vol. 29, no. 8, pp. 1470–1475, 2007.

[100] Vlasic, D., Baran, I., Matusik, W., and Popovic, J., "Articulated mesh animation from multi-view silhouettes," in *ACM SIGGRAPH*, 2008.

[101] Wang, J., Bhat, P., Colburn, R. A., Agrawala, M., and Cohen, M. F., "Interactive video cutout," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, (New York, NY, USA), pp. 585–594, ACM, 2005.

[102] Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., and Debevec, P., "Performance relighting and reflectance transformation with time-multiplexed illumination," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 756–764, 2005.

[103] Yersin, B., Maïm, J., Pettré, J., and Thalmann, D., "Crowd patches: populating large-scale virtual environments for real-time applications," in *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, (New York, NY, USA), pp. 207–214, ACM, 2009.

[104] Zhang, Z., "Flexible camera calibration by viewing a plane from unknown orientations," *Computer Vision, IEEE International Conference on*, vol. 1, p. 666, 1999.

[105] Zhao, L. and Safonova, A., "Achieving good connectivity in motion graphs," in *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*, 2008.

[106] Zhaozheng, Y. and Collins, R., "Shape constrained figure-ground segmentation and tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2009.

[107] Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R., "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600–608, 2004.

[108] Zivotofsky, A. and Hausdorff, J., "The sensory feedback mechanisms enabling couples to walk synchronously: An initial investigation," *Journal of NeuroEngineering and Rehabilitation*, vol. 4, no. 1, pp. 4–28, 2007.